(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

THE DEVELOPMENT OF A MODEL BUILDER FOR A MICROCIRCUIT SUBSTRATE

by

Patric Karl Roesch

June 1991

Thesis Advisor: A.D. Kraus

Approved for public release; distribution unlimited.

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings
2a Security Classification Authority 2b Declassification/Downgrading Schedule	3 Distribution Availability of Report Approved for public release; distribution is unlimited.
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School 6b Office Symbol *(If Applicable)* 32	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization 8b Office Symbol *(If Applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*	10 Source of Funding Numbers Program Element Number / Project No / Task No / Work Unit Accession No

11 Title *(Include Security Classification)* THE DEVELOPMENT OF A MODEL BUILDER FOR A MICROCIRCUIT SUBSTRATE

12 Personal Author(s) Patric K. Roesch

13a Type of Report	13b Time Covered	14 Date of Report *(year, month,day)*	15 Page Count
Master's Thesis	From	1991 , May 7	101

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	Model Builder; Development; Microcircuit; Substrate

19 Abstract *(continue on reverse if necessary and identify by block number*

The Naval Postgraduate School is currently in possession of software designed to perform a thermal analysis of electronic components. This software package incorporates a model builder whose primary function is to generate a thermal model. In its present configuration, the model builder requires an inordinate amount of time for data input and model verification. This thesis describes the development of a model builder designed specifically to reduce the time required to model the substrate, epoxy, and carrier layers of a microcircuit assembly.

20 Distribution/Availability of Abstract [X] unclassified/unlimited [ ] same as report [ ] DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual Allan D. Kraus	22b Telephone *(Include Area code)* (408) 646-2730	22c Office Symbol EC/KS

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted security classification of this page

All other editions are obsolete Unclassified

Approved for public release; distribution is unlimited.

The Development of a Model Builder for a
Microcircuit Substrate

by

Patric K. Roesch
Lieutenant, United States Navy
B.S., University of Florida, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1991

Author: Pat K Roesch

Patric Karl Roesch

Approved by: AD Kraus

A.D. Kraus, Thesis Advisor

Sherif Michael

S. Michael, Second Reader

M. A. Morgan

Michael A. Morgan, Chairman, Department of Electrical and
Computer Engineering

ABSTRACT

The Naval Postgraduate School is currently in possession of software designed to perform a thermal analysis of electronic components. This software package incorporates a model builder which contains two programs whose primary function is to generate a thermal model. In its present configuration, the model builder requires an inordinate amount of time for data input and model verification. This thesis describes the development of a model builder designed specifically to reduce the time required to model the substrate,epoxy and carrier layers of a microcircuit assembly.

TABLE OF CONTENTS

I. INTRODUCTION

The Naval Postgraduate School is currently in possession of software designed to perform a thermal analysis of electronic components. This software package incorporates a model builder which contains two programs whose primary function is to generate a thermal model, or input data file, to be read by the thermal analyzer program. The first program is considered to be a general model builder which is used in all model development stages as well as to modify an existing model. The second alternative was developed to generate a thermal model of a specific microcircuit geometry.

The development of an accurate thermal model of an electrical component requires that the structure be subdivided into a large number of small but finite subvolumes. Each subvolume is assumed to be isothermal with the centroids, also called nodes, considered to be representative of the entire subvolume. The most difficult problem encountered in the development of a thermal model is the generation of n-node equations in n-unknown temperatures where the nodes are connected by thermal conductances. As the desired accuracy of the thermal model increases, the number of required node equations becomes extremely large. Therefore, it is imperative

that the design engineer have access to a model builder that will produce the thermal model in a reasonable period of time.

In its present configuration, the thermal analysis software contains a model builder that generates the required node equations automatically. There is no question that the existing model builder programs have replaced the extremely laborious and time consuming process of generating the node equations by hand. However, they still require an inordinate amount of time for data input and model verification.

This thesis describes the development of a model builder designed specifically to reduce the time required to model the substrate, epoxy, and carrier layers of a microcircuit assembly. A typical microcircuit package configuration is shown in Figure 1. Figure 2a provides a horizontal interior illustration while Figure 2b displays the specific geometry to be modeled. All three layers may contain an equal number of nodes over their width. However, the carrier layer may contain a mounting ear on the front and rear surfaces. Additional characteristics to be discussed in what follows are:

1) The capability of working in English or SI units.

2) The choice of four aspect ratios.

3) The provision for up to 740 nodes depending on the existence of mounting surfaces (ears).

4) The ability to input heat dissipation using several methods.

5) The provision for six ambient temperatures.

6) The provision for rapid, menu-driven data input.

7) The automatic calculation of conductance values based on user input.

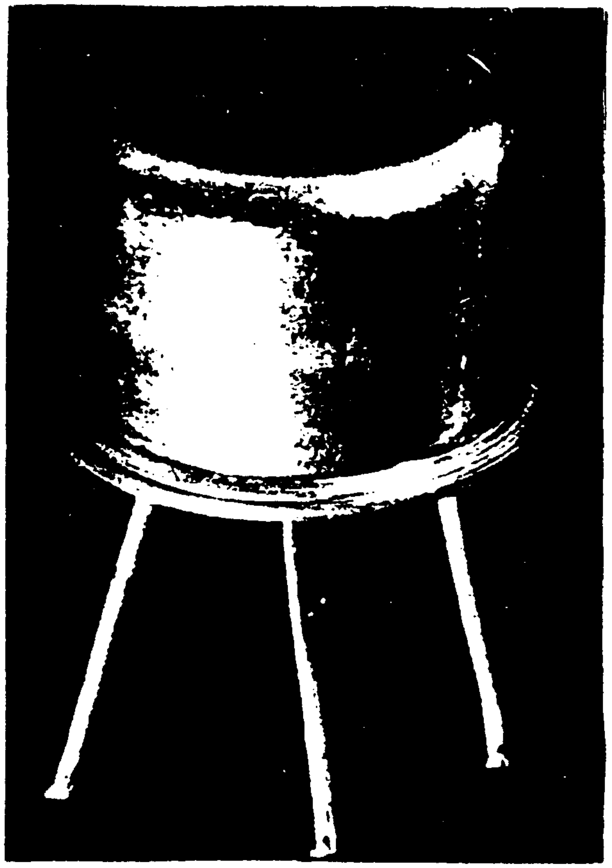
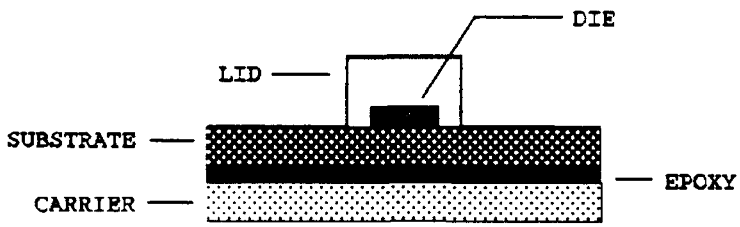
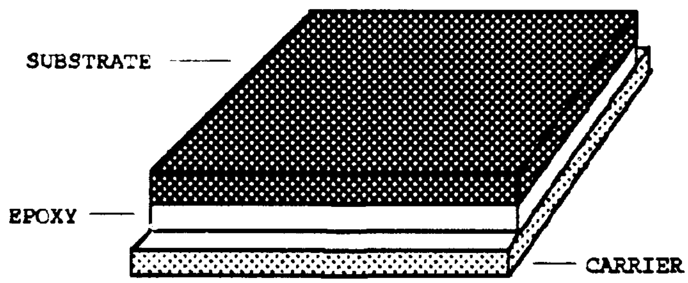


Figure 1. TO-5 configuration. (Courtesy of Honeywell,Inc.)



A. Horizontal cutout of a typical semiconductor assembly containing a single microcircuit die.



B. Area to be modeled.

Figure 2.

II. THE REASON FOR THERMAL ANALYSIS

Over the past several decades a trend of increasing sophistication and complexity has enveloped the electronics industry. This continuing advance in technology has greatly increased the reliability, capability, performance, and availability of electronic systems. The escalating demand for further advances in all areas of electronics has presented engineers with an abundance of complex problems.

One major area of concern is the continued development of advanced methods in the thermal control of multilayered structures. It is the responsibility of designers to ensure that electronic components operate efficiently and effectively throughout the specified thermal limits. Therefore, it is extremely important that design engineers have the capability to accurately and rapidly predict the temperature distribution on multilayered structures prior to prototype production. The overriding reasons for performing a precise thermal analysis are to increase component reliability, ensure proper material selection, ensure bias stabilization, and reduce or eliminate the possibility of catastrophic thermal failure.[Ref. 1]

A. RELIABILITY

There is a predictable relationship between the operating temperature of electronic components and reliability [Ref. 2]. The materials used in the fabrication of components have temperature limitations. Should these temperature boundaries be exceeded, the physical and chemical properties of the material are altered and the device fails. Figure 3 displays the intimate relationship between failure rate and component operating temperature for some selected devices. Furthermore, it is an established fact that the reliability of an electronic component is inversely proportional to the junction or component temperature and is also directly linked to failure rates [Ref. 1].

Consider Figure 4 which illustrates the "bathtub" mortality curve with the failure rate of a particular component plotted against component age during operation within thermal limits. The high failure rate in the interval prior to t_b, also known as the burn-in period, is considered to be the result of poor quality control during the fabrication process.[Ref. 1]

The area of highest concern is the interval between t_b and t_w. This period is considered to be the useful life, since with proper quality control, testing and burn-in procedures, t_b is equal to zero. Failures that occur in this interval are due to a variety of causes and are unpredictable.[Ref. 1]

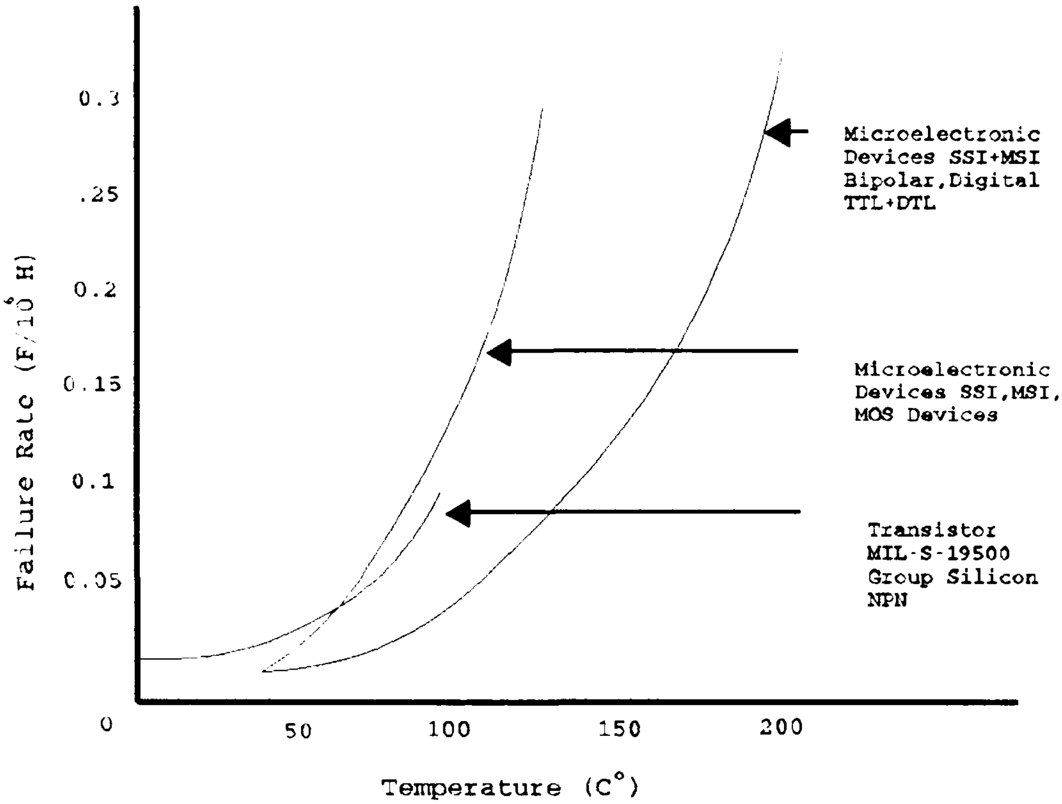


Figure 3. Failure rate vs. Temperature for selecte devices [Ref. 2].

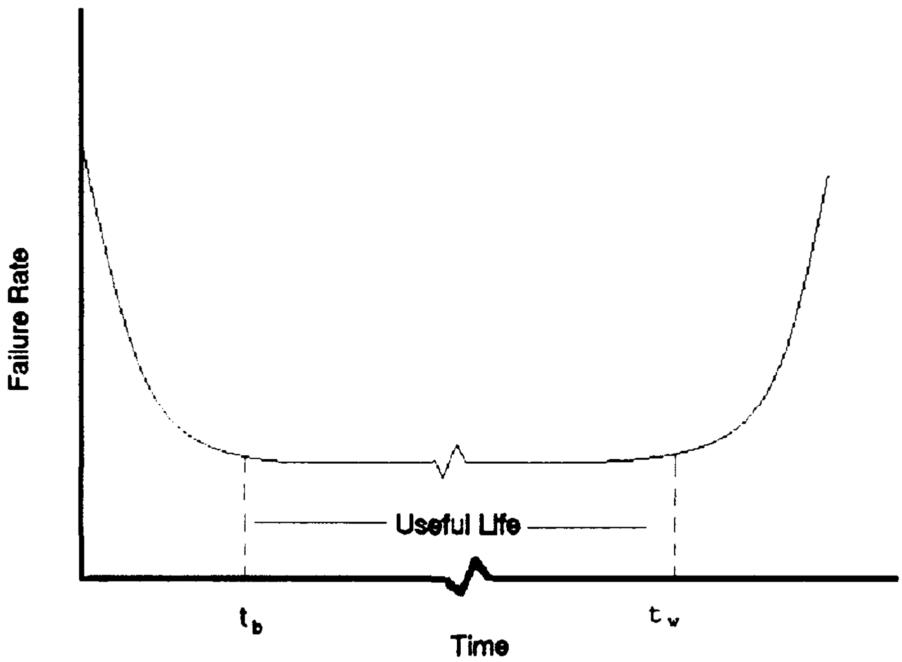


Figure 4. "Bathtub" mortality curve showing variation of failure rate with component age [Ref. 1].

Failure also occurs gradually due to sustained operations within specified temperature limits. If all aspects of fabrication have been performed correctly only a small fraction of components will have failed prior to reaching t_w, also known as the wear-out period. Failures that occur in this period are due to the slow and never ending deterioration of materials.

B. MATERIAL SELECTION

The fabrication of electronic components results in the joining of several different materials. Consider Figure 2 which depicts a typical semiconductor structure. When power is applied to the components, heat is dissipated to the substrate and subsequently to the carrier. The mechanical properties of these materials are all affected differently by changes in temperature. Opposing mechanical and chemical reactions due to environmental conditions and contaminants may result in component performance degradation or a reduction in useful life. Table 1 lists temperature related factors that may affect component performance.

The primary objective in the selection of materials for the fabrication of an electronic assembly is to achieve the desired level of correlation within the finished product. As packaging densities increase thermal, mechanical, electrical, and chemical coupling becomes very strong. This high level of coupling can be both an advantage and a disadvantage. For

example, a high level of correlation is desirable during fabrication to ensure an uniform product. However, in use, strong coupling is generally more desirable for moderate temperature deviations and weaker coupling is more desirable for large temperature deviations. In the case of large temperature deviations strong coupling may result in the catastrophic failure of many connected components while weak coupling may limit the number of failed components. Therefore, the strength of coupling between materials must be based on the type of failure most likely to occur and an accurate thermal analysis must supply this information to assist in proper material selection [Ref. 3].

TABLE 1. TEMPERATURE FACTORS [Ref. 4]

Mechanism	Effect on Equipment	Accelerating factors
Increasing Temperature	Loss of strength, reduced stiffness, reduced resonant frequency, softening, distortion, aging, and creep	Lubricants, rubber parts, plastics, corrosion, fatigue, load intensity, and time duration
Reducing Temperature	Increased viscosity, increased stiffness, increased resonant frequency, brittleness, and reduced impact resistance	Lubricants, rubber parts, plastics, and time duration
Thermal Expansion and Contraction	Change in size and shape, buckling, cracking, distortion, and loosening	Temperature cycling, temperature range, unequal expansion coefficients, stress concentrations , and lack of strain relief

C. BIAS STABILIZATION

The first step in the design and implementation of a semiconductor device is to establish a stable and predictable electrical operating point. This procedure, known as bias stabilization, attempts to determine a stable operating point that is virtually independent of external component parameters. However, as external parameters change, the operating point is directly affected. Therefore, a good bias design ensures that components will always operate within a certain range of their nominal value.[Ref. 5]

Consider Figure 5 which displays a transistor connected in the common-emitter configuration. Suppose that a proposed operation requires a specific collector to emitter voltage (V_{CE}). The circuit consists of a battery or some other source that provides a bias voltage V_{CC}, the collector resistor R_C, and the transistor. By Kirchoff's voltage law

$$-V_{CC} + R_C I_C + V_{CE} = 0 \qquad (1)$$

which results in a collector to emitter voltage of

$$V_{CE} = V_{CC} - I_C R_C \qquad (2)$$

Should the collector current be allowed to increase in excess of tolerable limits, V_{CE} must decrease because V_{CC} and R_C are

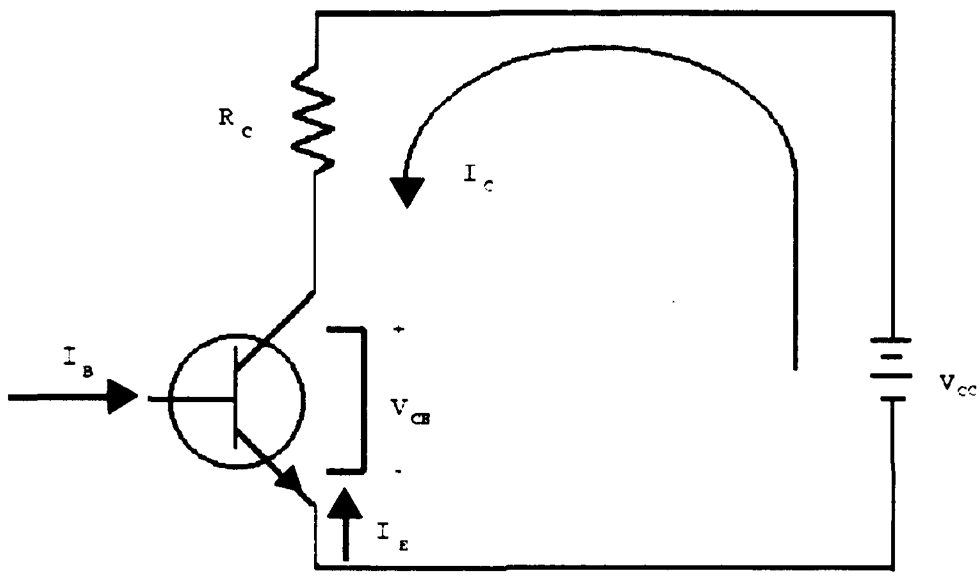


Figure 5. Schematic of transistor connected in common-emitter configuration [Ref. 1].

fixed values. Therefore, it should be noted that, if a high junction temperature causes an increase in I_C, V_{CE} can no longer be maintained at the level desired to perform the desired operation.[Ref. 1]

1. Operating in the Forward Bias Region

As an example of electronic component temperature dependence, consider a diode operating in the forward bias region. In the forward region the i-v relationship is closely approximated by

$$i = I_S(e^{\frac{v}{nV_T}} - 1) \tag{3}$$

In this equation I_S is a constant for a given diode at a given temperature. The current I_S is usually called the saturation current. However, another name for it is the scale current, which arises from the fact that I_S is directly proportional to the cross-sectional area of the diode. Furthermore, it can be seen in Table 2 that, I_S is a very strong function of temperature.[Ref. 5]

The temperature relationship between I_S and the forward current i is derived from the voltage V_T. This constant, called the thermal voltage, is given by

$$V_T = \frac{kT}{q} \tag{4}$$

where

k = Boltzman's constant, $1.38 \cdot 10^{-23}$ J/ K

T = the absolute temperature, K

q = the charge on the electron, $1.602 \cdot 10^{-19}$ C

Table 2 illustrates this relationship and emphasizes the need to accurately analyze a proposed assembly prior to fabrication.

TABLE 2. TEMPERATURE DEPENDENCY OF I_S ON i FOR SELECTED MATERIALS

GERMANIUM			SILICON	
°C	I_S	i	I_S	i
25	3.0 μA	18.01 μA	50.0 ηA	82.31 ηA
95	0.384 mA	1.473 mA	51.2 μA	61.39 μA
165	49.2 mA	0.136 A	52.4 mA	49.2 mA

D. CATASTROPHIC THERMAL FAILURE

Another of the primary goals of techniques in advanced thermal control is to provide a thermal environment for a diversity of components that are in increasingly close proximity to each other. Figure 6 illustrates the increasing level of packaging densities. With increasing complexity comes an increased level of connections between dissimilar material and a greater possibility for exceeding temperature limitations. Therefore, it is necessary that designers have

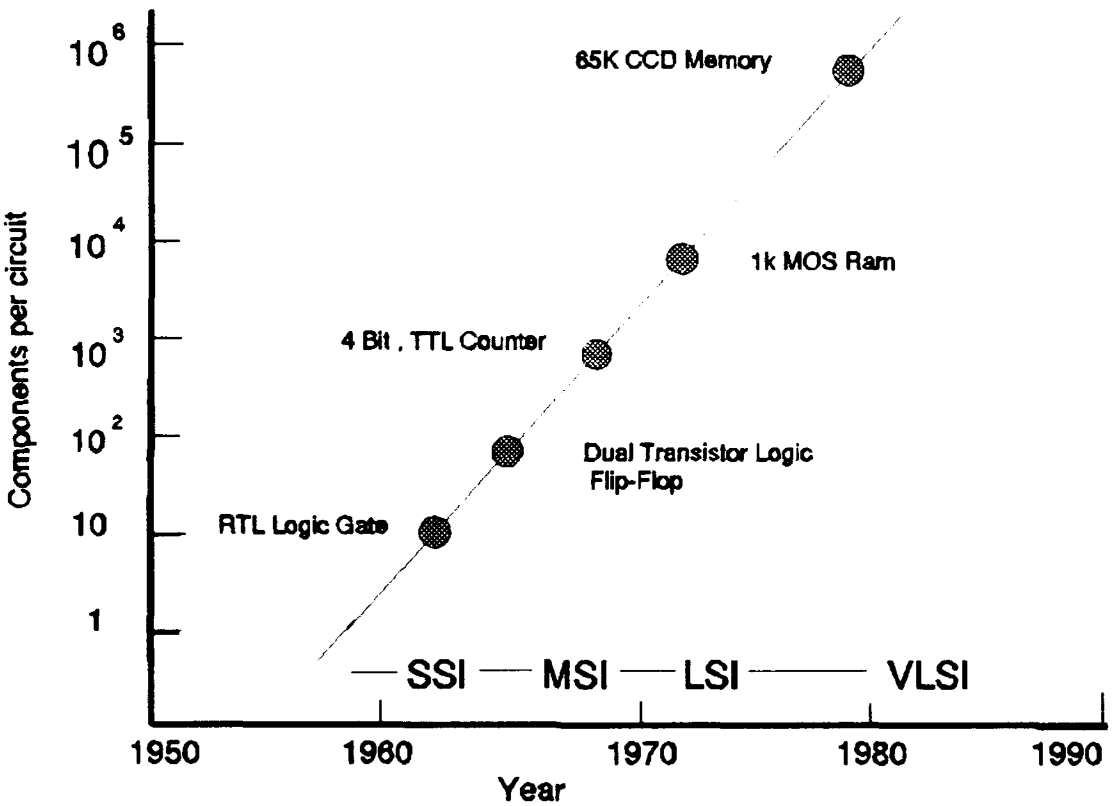


Figure 6. State of the art in circuit complexity [Ref. 1].

some knowledge of the possible range of component and environmental variations in order to prevent catastrophic thermal failure.[Ref. 1]

Catastrophic thermal failure is defined as an immediate, thermally induced, total loss of electronic function in a specified component. This type of failure is the result of a component melting due to excessive temperature, a thermal fracture of the substrate or carrier, or a separation of leads and the external network. It is generally considered to be dependent on the local temperature field, operating history, and operating modes of the component. As previously stated, a variety of problems arise when components are subjected to temperatures in excess of their rated limits. Furthermore, it is extremely difficult to determine the precise temperature at which catastrophic failure may occur. The incorporation of an accurate thermal analysis, in combination with test and operating experience, may be used to generate a catastrophe free upper operating limit. These maximum allowable operating temperatures are used to generate the master thermal control configuration for the system.[Ref.1]

III. HEAT TRANSFER

Heat transfer is defined as all energy flows that arise as a result of temperature differences [Ref. 6]. Because electronic components are not one hundred percent efficient, they produce heat as well as the desired output. In the case of semiconductor devices, heat develops in parts having low thermal efficiencies, such as the die. One of the major objectives of packaging is to develop an effective system for the removal of heat from these parts [Ref. 4]. It is imperative that design engineers understand all modes of heat transfer in order to incorporate an efficient method of heat removal into component designs. The modes of heat transfer are conduction, convection, and radiation.

A. CONDUCTION

Conduction is the transfer by molecular motion of heat between one part of a body to another part of the same body or between one body and another in physical contact [Ref. 1]. Joseph Fourier, a French physicist, proposed that the rate of heat flow through a material by conduction is proportional to the area of the material normal to the heat flow path and to the temperature gradient along the heat flow path.

This proportionality is represented mathematically by

$$q \propto -A\frac{dT}{dx} \tag{5}$$

where the minus sign allows for a positive heat flow in the presence of a negative temperature gradient. The introduction of a proportionality constant, known as thermal conductivity, results in the following rate equation which describes this mechanism [Ref. 1]:

$$q = -kA\frac{dT}{dx} \tag{6}$$

where

k = thermal conductivity of the material, W/m-°C

A = area of the heat flow path, m^2

dT/dx = change in temperature per unit length, °C/m

q = rate of heat flow, W

1. General Equation of Heat Conduction

The first step in the analytical solution of a heat conduction problem for a given structure is to choose an orthogonal coordinate system such that the surfaces coincide with the boundary surfaces of the structure [Ref. 7]. In the case of the model builder developed in this thesis,the

rectangular coordinate system will be employed. The general equation of heat conduction is given as

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) + q_i = \rho C\frac{\partial T}{\partial t} \tag{7}$$

Then, assuming k, C, and ρ are independent of temperature, direction, and time, the resulting equation is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{q}{k} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{8}$$

where

T = temperature, C

x, y, and z = cartesian coordinates, m

t = time, sec

k = thermal conductivity, W/m- C

q = internal heat generation, W/m^3

α = thermal diffusivity, $k/\rho C$, m^2/sec

There are several variations of the general equation of conduction. The first, known as the Fourier equation, provides a solution for a system that contains no heat sources:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{9}$$

The second variation, known as the Poisson equation, supplies a solution for a system in which the temperature does not vary with time:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{q}{k} = 0 \qquad \textbf{(10)}$$

The third and final variation of the general equation of conduction provides a solution for a system void of heat sources and operating in steady state. The resulting equation, known as the Laplace equation, is given as

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \nabla^2 T = 0 \qquad \textbf{(11)}$$

2. Simple Plane Slab

Consider Figure 7 which illustrates a simple plane slab with face temperatures T_1 and T_2. Using only one dimension, equation (11) is reduced to

$$\frac{d^2 T}{dx^2} = 0$$

By integrating twice and applying boundary conditions the temperature distribution across the slab is seen to be

$$T = T_1 - \frac{x}{L}(T_1 - T_2) \qquad \textbf{(12)}$$

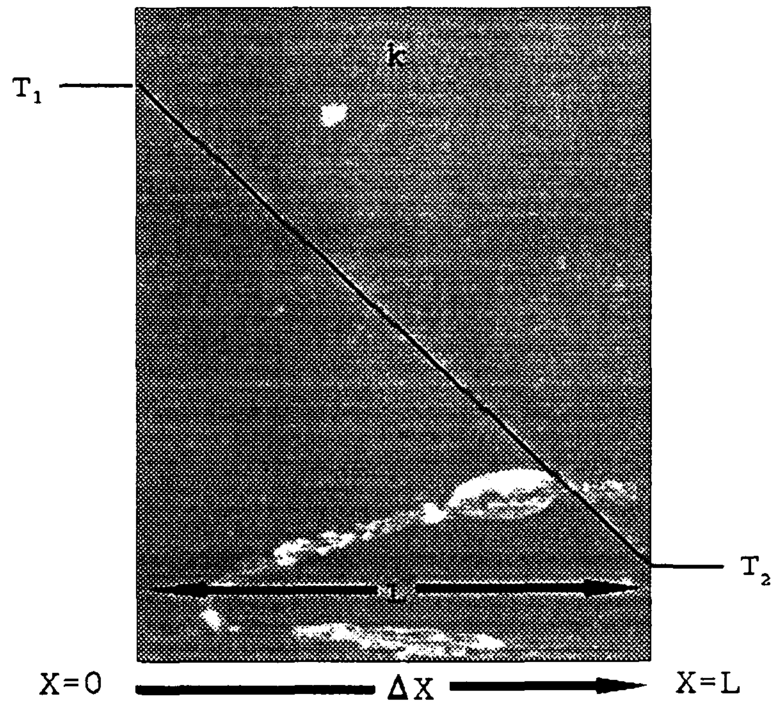


Figure 7. Conduction heat transfer through a simple plane slab.

Insertion of equation (12) into equation (6) produces a solution for the heat flow across the slab:

$$q = -kA\left[\frac{-(T_1 - T_2)}{L}\right] = \frac{kA}{L}(T_1 - T_2) \tag{13}$$

Ohm's Law indicates a direct analogy between heat flow, equation (13), and the flow of electrical current through a resistor, V = RI. This electrothermal analogy is extremely useful in the solution of one dimensional, steady state problems without energy generation and will be developed further in the next section.[Ref. 8]

3. Electrothermal Analog

As previously stated, there is a direct analogy between heat flow across a simple plane slab, equation (13), and electrical current governed by Ohm's law:

$$I = \frac{V}{R} \tag{14}$$

In this case, the analogous quantities are

Current I $\leftrightarrow$ Heat Flow q

Potential V $\leftrightarrow$ Temperature Difference ΔT

Resistance R $\leftrightarrow$ Thermal Resistance R

It is easily seen that for the heat flow in a simple plane slab described by, equation (13), the thermal resistance is

$$R = \frac{\Delta T}{q} = \frac{L}{kA} \tag{15}$$

The electrothermal analog for conduction across a simple plane slab is shown in Figure 8.[Ref. 1]

B. CONVECTION

Convection is defined as the process by which thermal energy is transferred to or from a solid by a fluid flowing past it. Should the fluid flow be the result of a temperature difference the phenomena is called natural or free convection. On the other hand, when a pump or fan causes the mass movement the process is called forced convection.[Ref. 1]

Recall that at the interface between a solid and a fluid that heat is transferred by conduction and must obey Fourier's law, equation (6). Due to the difficulty encountered in accurately measuring the temperature gradient, Newton suggested that the surface heat transfer rate be related to the product of surface area and the temperature difference between the surface and the fluid. The results of this proposition lead to Newton's law of cooling:

$$q = hA(T_0 - T_f) \tag{16}$$

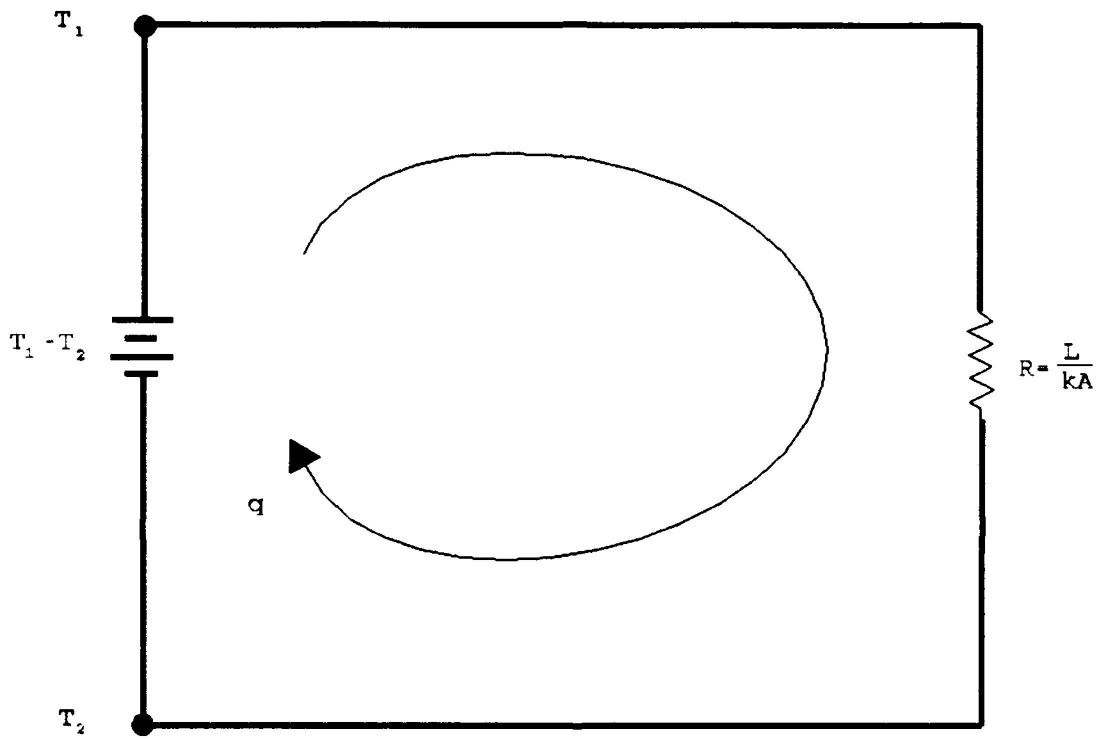


Figure 8. Conduction Electrothermal Analog.

where h is a proportionality factor that has become known as the surface heat transfer coefficient.[Ref. 1]

From a comparison of Newton's law of cooling, equation (16), and Fourier's law, equation (6), one can derive that the surface heat transfer coefficient can be related to the thermal conductivity, the wall temperature gradient of the fluid, and the surface fluid temperature difference:

$$h = \frac{q}{A \cdot \Delta T} = \frac{-k\left(\frac{\partial T}{\partial y}\right)}{\Delta T} \quad \textbf{(17)}$$

Therefore, any correlation between heat transfer coefficients must reflect the dependence of h on the thermal conductivity of the fluid and on the ratio of the wall temperature gradient to the temperature difference.[Ref. 1]

1. Electrothermal Analog

The addition of heat transfer by convection to both surfaces of the simple plane slab of Figure 7 results in the configuration shown in Figure 9. In the convective case thermal resistance is represented by

$$R = \frac{1}{hA} \quad \textbf{(18)}$$

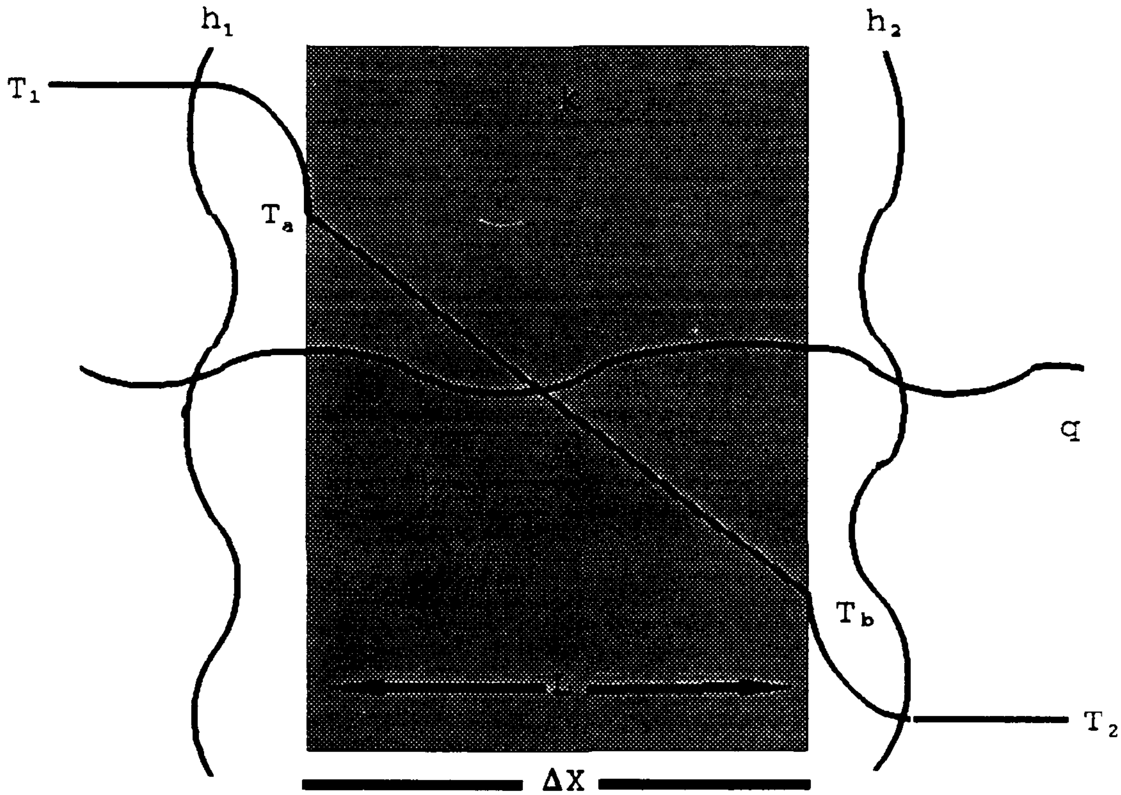


Figure 9. Convective heat transfer on a simple plane slab.

Therefore, the total thermal resistance is

$$R = \frac{1}{h_1 A} + \frac{L}{kA} + \frac{1}{h_2 A} = \frac{1}{A}\left[\frac{1}{h_1} + \frac{L}{k} + \frac{1}{h_2}\right] \tag{19}$$

which is represented by the electrothermal analog shown in Figure 10. A simple consideration of circuit theory then shows that:

$$q = \frac{\Delta T}{R} = \frac{(T_1 - T_2)}{\frac{1}{A}\left[\frac{1}{h_1} + \frac{L}{k} + \frac{1}{h_2}\right]} \tag{20}$$

C. RADIATION

Heat transfer by radiation is the means by which thermal energy can be transmitted through a space without an intervening medium while obeying the laws of electromagnetics. Thermal radiation, while traveling at the speed of light, may be absorbed, reflected, or transmitted upon contact with a surface. An ideal black body absorbs all incident radiation and reflects and transmits none of it. The concept of the black body is useful because laws governing its radiation are simple and many real bodies may be treated approximately as black bodies [Ref. 1].

Materials used in the fabrication of electronic components are classified as gray. Gray bodies are diffusely reflecting

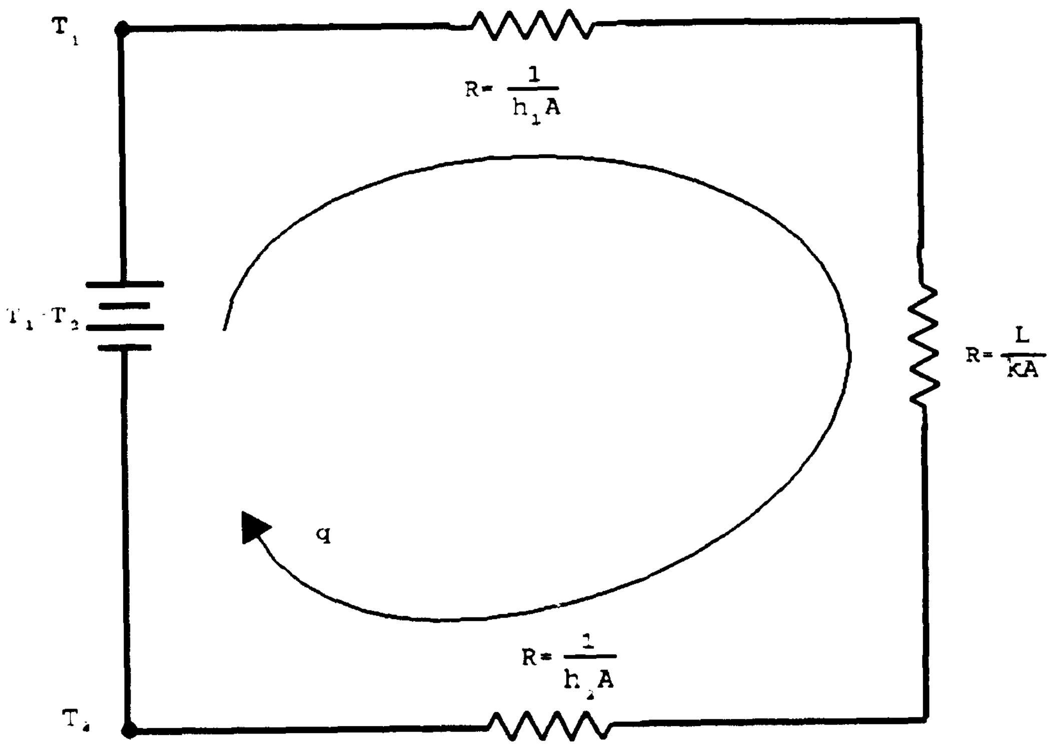


Figure 10. Conduction and convection electrothermal equivalent.

opaque surfaces [Ref. 4]. These surfaces reflect equal amounts of energy over the thermal radiation spectrum (wavelengths of about 0.1 μm to about 100 μm) in all directions. The heat transfer efficiency of this mode depends on the configuration, the orientation, and the temperatures of the surfaces in the electronic assembly.

1. Transformation of the General Radiation Equation

The use of the thermal radiation equation in analytical studies is made difficult by its dependence on the fourth power relationship between the temperatures. Due to the nonlinear characteristics and the complexity of the calculations a computer program is the desired method to solve problems that have a significant transfer by radiation. The general equation for radiation interchange is:

$$q = \sigma F_A F_\epsilon A \ (T_s^4 - T_r^4) \tag{21}$$

where

σ = Stefan-Boltzman constant, $5.669 \cdot 10^{-8}$ W/m^2- K^4

F_A = shape factor that accounts for the arrangement of the of the radiating source and absorbing receiver

F_ϵ = emissivity factor that accounts for the ability of the source and receiver to emit or absorb radiation

T_s = temperature of the source, K

T_r = temperature of the receiver, K

A = surface area, m^2

It is important to note that the absolute temperature scale must be used when considering radiation.

One way to handle computations involving heat transfer by radiation is to transform the general radiation equation, equation (21), into a configuration compatible with Fourier's law. In this case, linearization of the general radiation equation is the method used to produce the desired result. This is achieved by factoring the difference in the fourth power of the temperatures as follows:

$$(T_s^4 - T_r^4) = (T_s^2 + T_r^2)(T_s^2 - T_r^2)$$
$$= (T_s^2 + T_r^2)(T_s + T_r)(T_s - T_r) \tag{22}$$

Inserting this into equation (21) results in

$$q = \sigma F_A F_e A \; (T_s^2 + T_r^2)(T_s + T_r)(T_s - T_r) \tag{23}$$

A radiative heat transfer coefficient may therefore be defined as

$$h_r = \sigma F_A F_E (T_S^2 + T_R^2)(T_S + T_R) \tag{24a}$$

or

$$h_r = \sigma F_A F_E (T_S^3 + T_S^2 T_R + T_S T_R^2 + T_R^3) \tag{24b}$$

Then, substituting h_r into equation (23), radiation heat transfer may be treated exactly as convection at the boundary.

A thermal resistance for radiation heat transfer can now be proposed:

$$R = \frac{1}{h_r A} \tag{25}$$

Figure 11 provides an illustration of the electrothermal equivalent with the addition of radiation resistance in parallel with convective resistance.

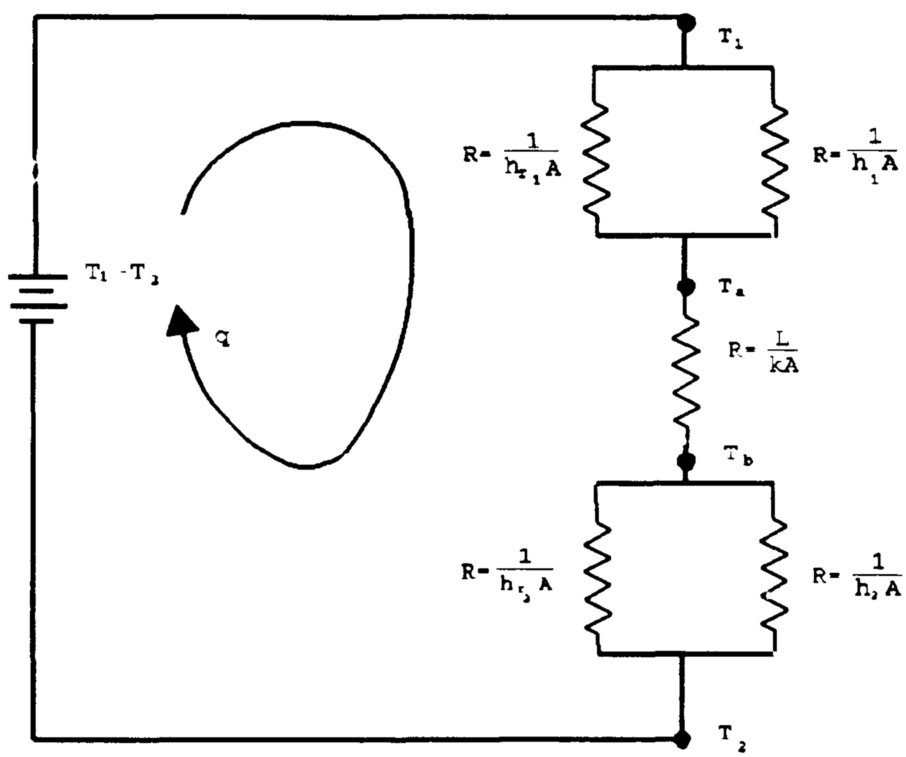


Figure 11. Radiation, convection, and conduction electrothermal equivalent.

2. General Problems of Heat Transfer by Radiation

In the calculation of heat transfer by radiation, it is usually necessary to approximate real-body behavior by the gray-body idealization. The assumption that emissivity is always equal to absorptivity is frequently required if the problem is to be solved. Even a simple situation becomes quite complex if real body behavior is considered and the resulting convenience of replacing absorptivity with emissivity is lost. A second difficulty in considering real-body behavior is the lack of sufficient data. To properly account for real-body behavior, extensive tabulations would be required.[Ref. 6]

The usual problem of heat transfer by radiation is further complicated not only because the surfaces are nonblack but also because the configuration of the areas involved is not simple and reflecting surfaces may be present to augment the direct exchange [Ref. 6]. There are a number of methods available to assist in the solution of heat transfer by radiation, however, their development is beyond the scope of this thesis.

IV. FINITE DIFFERENCES

The need to utilize a computer to determine the temperature distribution within an electronic component or system has made finite difference methods very desirable. This numerical method is useful in problems involving nonlinearities, complex geometries, complicated boundary conditions, or a system of coupled partial differential equations. The purpose of this section is to provide the reader with some basic concepts involved in finite difference methods for solving differential equations. Furthermore, it demonstrates the methodology used to formulate n-node equations in n-unknown temperatures and instills confidence that this numerical method is capable of generating an accurate thermal model.

A. FUNDAMENTAL CONCEPTS

Consider equation (7), the general equation for heat transfer by conduction. In order to produce a numerical solution to a conduction heat transfer problem it is necessary to reconfigure the partial differential equation into a form that allows differentiation to be performed by numerical methods. Therefore, it is essential that accurate approximations of the first and second derivatives be obtained.

1. First Derivative Approximation

The derivative of a function at any point can be expressed by a finite difference approximation by incorporating a Taylor series expansion about that point. Consider Figure 12 where T(x) is a function that can be expanded by a Taylor series. The Taylor series expansions of the functions T(x+Δx) and T(x-Δx) about a point x are:

$$T(x+\Delta x) = T(x) + \Delta x T'(x) + \frac{(\Delta x)^2}{2!}T''(x) + \frac{(\Delta x)^3}{3!}T'''(x) + \ldots \qquad \textbf{(26a)}$$

$$T(x-\Delta x) = T(x) - \Delta x T'(x) + \frac{(\Delta x)^2}{2!}T''(x) - \frac{(\Delta x)^3}{3!}T'''(x) + \ldots \qquad \textbf{(26b)}$$

In order to determine the first derivative, equations (26a) and (26b) are solved for T'(x).

$$T'(x) = \frac{T(x+\Delta x) - T(x)}{\Delta x} - \frac{(\Delta x)}{2}T''(x) - \frac{(\Delta x)^2}{6}T'''(x) - \ldots \qquad \textbf{(27)}$$

$$T'(x) = \frac{T(x) - T(x-\Delta x)}{\Delta x} + \frac{(\Delta x)}{2}T''(x) - \frac{(\Delta x)^2}{6}T'''(x) + \ldots \qquad \textbf{(28)}$$

Subtracting equation (26b) from (26a) and solving for T'(x) produces

$$T'(x) = \frac{T(x+\Delta x) - T(x-\Delta x)}{2\Delta x} - \frac{(\Delta x)^2}{6}T'''(x) - \ldots \qquad \textbf{(29)}$$

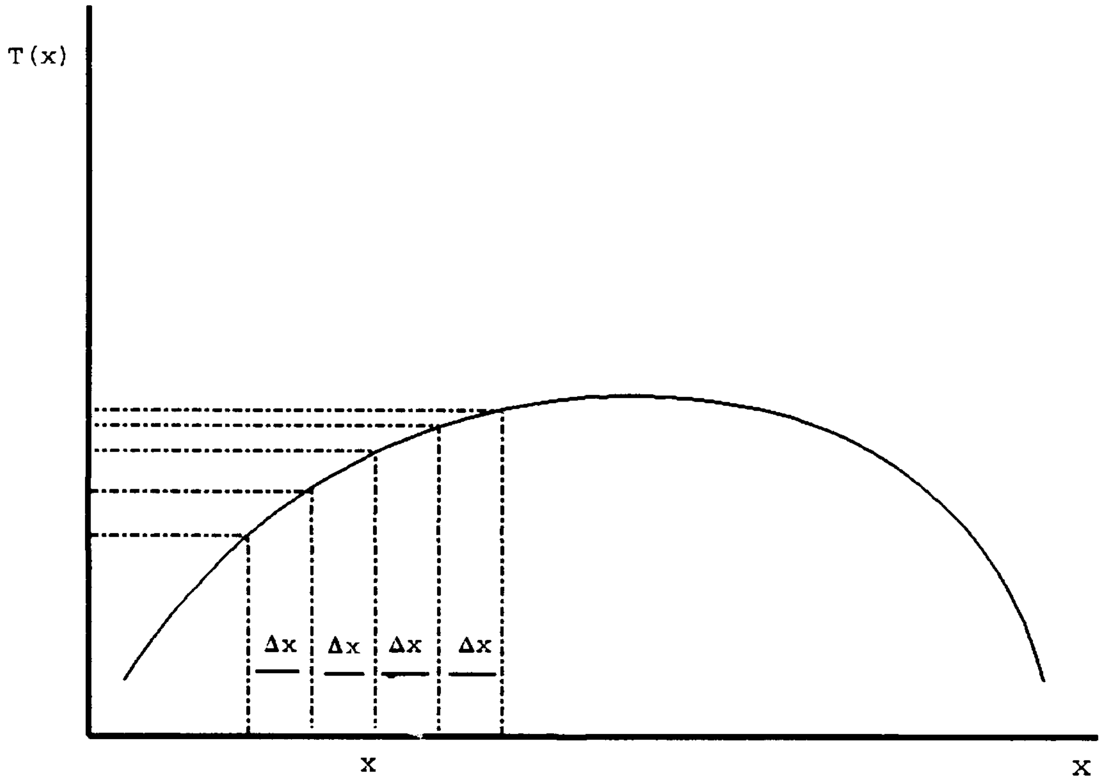


Figure 12. First and second derivative approximations.

From equations (27), (28), and (29) the first derivative approximations of T′(x) are determined to be

$$T'(x) = \frac{dT}{dx} = \frac{T(x+\Delta x) - T(x)}{\Delta x} \quad \textit{forward difference} \tag{30}$$

$$T'(x) = \frac{dT}{dx} = \frac{T(x) - T(x-\Delta x)}{\Delta x} \quad \textit{backward difference} \tag{31}$$

$$T'(x) = \frac{dT}{dx} = \frac{T(x+\Delta x) - T(x-\Delta x)}{2\Delta x} \quad \textit{central difference} \tag{32}$$

and for all it is observed that

$$T'(x) = \frac{dT}{dx} = \frac{\Delta T}{\Delta x}$$

2. Second Derivative Approximation

To obtain the second derivative of the function T(x), Figure 12, consider the Taylor series expansions of the functions T(x+2Δx) and T(x−2Δx) about a point *x*.

$$T(x+2\Delta x) = T(x) + 2\Delta x T'(x) + 2(\Delta x)^2 T''(x) + \frac{4}{3}(\Delta x)^3 T'''(x) + \ldots \tag{33a}$$

$$T(x-2\Delta x) = T(x) - 2\Delta x T'(x) + 2(\Delta x)^2 T''(x) - \frac{4}{3}(\Delta x)^3 T'''(x) + \ldots \tag{33b}$$

Inserting the corresponding first derivative approximation into equations (33a) and (33b) and solving for $T''(x)$ results in

$$T''(x) = \frac{d^2T}{dx^2} = \frac{T(x) + T(x+2\Delta x) - 2T(x+\Delta x)}{(\Delta x)^2} \quad \textit{forward difference} \tag{34}$$

$$T''(x) = \frac{d^2T}{dx^2} = \frac{T(x-2\Delta x) + T(x) - 2T(x-\Delta x)}{(\Delta x)^2} \quad \textit{backward difference} \tag{35}$$

The central difference is obtained by eliminating $T'(x)$ between equations (26a) and (26b).

$$T''(x) = \frac{d^2T}{dx^2} = \frac{T(x-\Delta x) + T(x+\Delta x) - 2T(x)}{(\Delta x)^2} \quad \textit{central difference} \tag{36}$$

and for all it is observed that

$$T''(x) = \frac{d^2T}{dx^2} = \frac{(\Delta T)}{(\Delta x)^2}$$

B. NODE ANALYSIS

The first step in the physical formulation of a solution by node analysis is, as previously stated, to divide the region into a finite number of subvolumes as shown in Figure 13. The centroid of each subvolume is called a node and is considered to be representative of the entire subvolume.

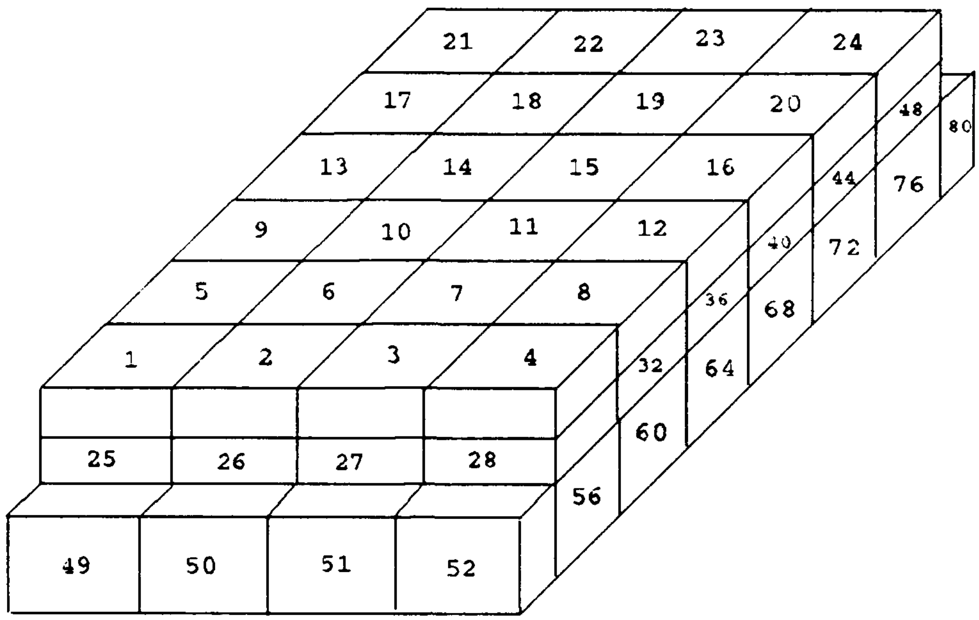


Figure 13. Structure divided into subvolumes.

At this point, continuity must be applied. In the steady state, the algebraic sum of all the heat leaving any node must equal zero. As an example, consider an interior node on the upper layer; select node number 14. Figure 14 provides an expanded picture around node-14. In order to maintain clarity the external environment will be considered as a single node, number 81. The resulting energy balance or node equation can be written as

$$q_{14,10} + q_{14,13} + q_{14,15} + q_{14,18} + q_{14,38} + q_{14,81} + q_i = 0 \quad \textbf{(37)}$$

Here, each numerical subscript represents the heat flow from node-14 to the indicated node. Furthermore, q_i allows for energy input by an external source such as a dissipating component.

The energy balance simply supplies the pertinent energy terms. It is important to note that differences in thicknesses and thermal conductivities must be taken into consideration. The rate equations may be written in terms of thermal resistances and temperature differences:

$$q_{14,13} = \frac{k_1 \Delta x \Delta z_1}{\Delta y} (T_{14} - T_{13}) \quad \textbf{(38a)}$$

$$q_{14,15} = \frac{k_1 \Delta x \Delta z_1}{\Delta y} (T_{14} - T_{15}) \quad \textbf{(38b)}$$

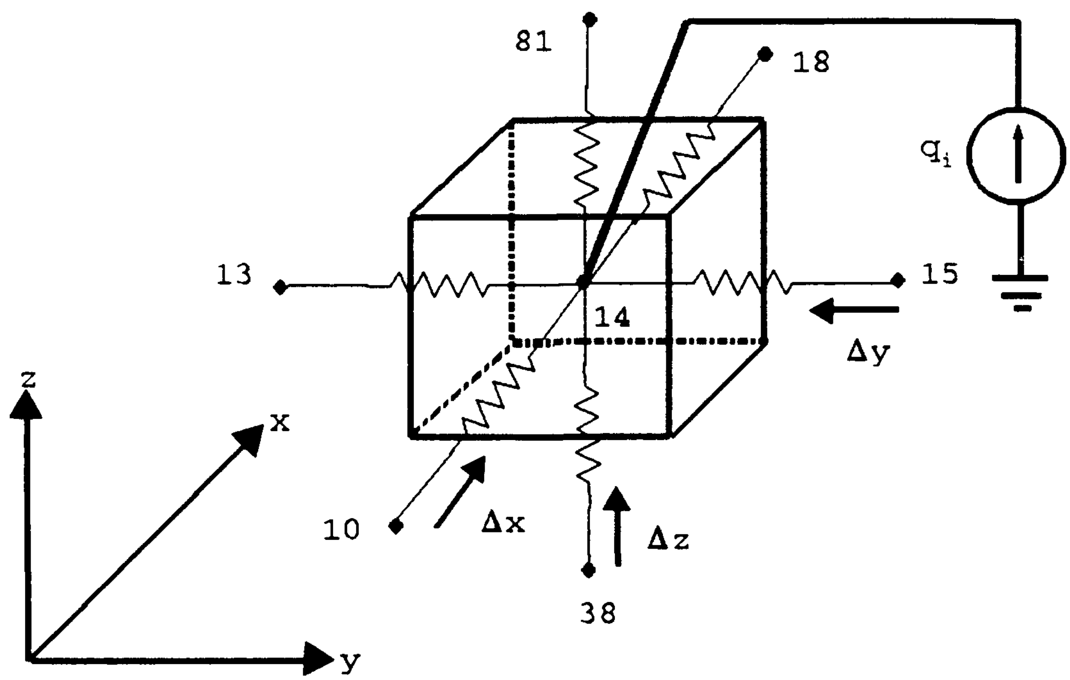


Figure 14.

$$q_{14,10} = \frac{k_1 \Delta y \Delta z_1}{\Delta x} (T_{14} - T_{10}) \tag{38c}$$

$$q_{14,18} = \frac{k_1 \Delta y \Delta z_1}{\Delta x} (T_{14} - T_{18}) \tag{38d}$$

$$q_{14,81} = \frac{2 k_1 \Delta x \Delta y}{\Delta z_1} (T_{14} - T_{81}) \tag{38e}$$

$$q_{14,38} = \frac{2 \Delta x \Delta y}{\left(\frac{\Delta z_1}{k_1}\right) + \left(\frac{\Delta z_2}{k_2}\right)} (T_{14} - T_{38}) \tag{38f}$$

where

k_1 and k_2 = thermal conductivities of the top layer and middle layer, respectively, W/m- C

Δx and Δy = path lengths in their respective direction and are node dimensions for surface area calculations, m

Δz_1 and Δz_2 = the thicknesses for the upper and middle layers, respectively, m

T_x = temperature of the node indicated by the subscript, °C

Substituting these rate equations into the energy balance equation, equation (37), results in an equation containing five unknown temperatures.

For simplicity let $\Delta x=\Delta y=\Delta z_1=\Delta z_2=1$ and $k_1=k_2=k$. Then

$$k(T_{14} - T_{10}) + k(T_{14} - T_{13}) + k(T_{14} - T_{15}) + k(T_{14} - T_{18})$$

$$+ 2k(T_{14} - T_{81}) + k(T_{14} - T_{38}) + q_i = 0 \qquad \textbf{(39)}$$

further reduction results in

$$-T_{10} - T_{13} - T_{15} - T_{18} - 7T_{14} - T_{38} = \frac{q_i}{k} + 2T_{81} \qquad \textbf{(40)}$$

An equation of this type can be written for every node in the region whose temperature is unknown. This results in a set of n-equations that relate the n-unknown temperatures. There are a variety of methods available to solve for the unknown temperatures. In large structures the most desirable method would be a thermal analysis program.

V. THE MODEL BUILDER

As previously stated, the model builder currently being used by the thermal analysis software package requires an inordinate amount of time for data entry. This thesis presents a specific model builder, TMDL, that will become part of the software package. TMDL provides a user friendly, rapid method to assist in the development of a thermal model with a specific geometry as in Figure 2. Specifically, it generates a properly formatted output data file for use by the thermal analyzer enroute to preparing an accurate listing of the temperature distribution of the structure.

A. THE THERMAL ANALYZER INPUT DATA FILE

The model builder generates a data file from physical characteristics of the structure provided by the user. This data file, also called the thermal analyzer input data file, must be in a format that is completely acceptable to the thermal analyzer. Furthermore, it will consist of five lines and up to seven data sets. This section describes each line and data set and their relationship with TMDL. Figure 15 displays a partial data file.

Line one is the title line. It may be left blank or may contain up to 79 alphanumeric characters. The user selected title appears at the top of the data file.

```
        EXAMPLE OUTPUT DATA FILE
740      6        0        0        0        0        0        0        2
 0       0        0
750      50       6        2        4        6        0        0        0
0
.0500000  .6666700               12  .8000000  78.00000
76.500000000 76.900000000 75.900000000 77.100000000 77.400000000 75.6000
7  7551          21           7521           111           7511          2411
73.700       36.850       87.709        43.855         2.047         1.456
6    11          31           7521           121           7511          2421
36.850       36.850       87.709        43.855         2.047         1.456
7    21          41           7521           131           7511          2431
36.850       36.850       87.709        43.855         2.047         1.456
6    31          51           7521           141           7511          2441
36.850       36.850       87.709        43.855         2.047         1.456
6    41          61           7521           151           7511          2451
36.850       36.850       87.709        43.855         2.047         1.456
6    51          71           7521           161           7511          2461
36.850       36.850       87.709        43.855         2.047         1.456
7    61          81           7521           171           7511          2471
36.850       36.850       87.709        43.855         2.047         1.456
6    71          91           7521           181           7511          2481
36.850       36.850       87.709        43.855         2.047         1.456
6    81         101           7521           191           7511          2491
36.850       36.850       87.709        43.855         2.047         1.456
7    91        7541           7521           201           7511          2501
36.850       73.700       87.709        43.855         2.047         1.456
6  7551         121             11           211           7511          2511
73.700       36.850       43.855        43.855         2.047         1.456
6   111         131             21           221           7511          2521
36.850       36.850       43.855        43.855         2.047         1.456
7   121         141             31           231           7511          2531
36.850       36.850       43.855        43.855         2.047         1.456
6   131         151             41           241           7511          2541
36.850       36.850       43.855        43.855         2.047         1.456
6   141         161             51           251           7511          2551
36.850       36.850       43.855        43.855         2.047         1.456
6   151         171             61           261           7511          2561
36.850       36.850       43.855        43.855         2.047         1.456
6   161         181             71           271           7511          2571
36.850       36.850       43.855        43.855         2.047         1.456
6   171         191             81           281           7511          2581
36.850       36.850       43.855        43.855         2.047         1.456
7   181         201             91           291           7511          2591
36.850       36.850       43.855        43.855         2.047         1.456
6   191        7541            101           301           7511          2601
36.850       73.700       43.855        43.855         2.047         1.456
6  7551         221            111           311           7511          2611
73.700       36.850       43.855        43.855         2.047         1.456
6   211         231            121           321           7511          2621
36.850       36.850       43.855        43.855         2.047         1.456
```

Figure 15. Output data file.

Line two is the problem data line. It has nine entries of which two are under user control, number of nodes under consideration and unit type. One entry, the number of constant temperatures is preset at six for this specific model. The remaining entries have applications to models associated with heaters, unique exponents, secondary heat input, temperature coefficients and curves, and nodes controlling fast heat. These entries are not applicable to this model and are preset to zero.

Line three places a zero at three points and is beyond the user's control. Therefore, no further discussion is required.

Line four is the problem capability line. This line defines the maximum values for the entries in line two. The first entry is 750, the number of nodes for which the analysis is dimensioned. This is significant because the first constant temperature will be assigned the node number 751. The second entry is 50 which is the maximum number of constant temperatures in accordance with the analyzer dimension statement. Therefore, it is possible to have 50 constant temperatures allocated from node 751 to 800. The third entry is preset to six for an application concerning heaters and is not applicable to this model. The balance of the entries in line four represent a listing of data sets that are required for the particular analysis at hand. TMDL uses three data sets that will be discussed in what follows.

Line five contains five items that concern the level of accuracy that the thermal analyzer will achieve. These entries are preset. Item one provides the desired level of accuracy between iterations. Because the thermal analyzer solves iteratively, there must be an error criterion at which point calculations cease. A number that is too small will cause the computer to run excessively and a number that is too large will not give the desired accuracy. Therefore, a trade off is necessary and such a tradeoff has indicated that 0.05 is satisfactory. Item two is a damping factor that is used between iterations to prevent temperature oscillations between iterations. The third entry is the maximum number of iterations. This prevents excessive computer time in the event of faulty input data. Item four is a convergence factor that adjusts the damping to close to the critical value. This means that once the computer determines convergence is occurring, slow convergence is increased to reduce computer time. The fifth and final entry in line five is the initial temperature at which the iterative process begins. It is input by the user.

Input data set one contains temperature dependent coefficients and is not used in this model.

Input data set two contains up to 50 constant temperature inputs. This model has six constant ambient temperatures input by the user.

Input data set three involves heaters for fast warm up and is not used in this model.

Input data set four contains all pertinent information concerning the n-node equations. Each node requires two lines of data. The odd numbered lines are used for specifying the nodes that interact with the node in question and the modes by which this interaction takes place. Consider Figure 14, line one of data set four is as follows;

6 7551 21 7521 111 7511 2411

This line of data is for node number one of aspect ratio selection one. The first entry provides the number of connections to that node. Entry two says node 755, an ambient temperature, is connected to the node in question and the one indicates that the connection is by conduction. The rest of the entries are read similarly with numbers between 7511 and 7561 indicating a conductive connection with the external environment. External heat input is defined by entry 9991.

The even lines are used for specifying the inter node conductance values of the corresponding entry in the previous line.

Input data set five is used only if there are unique exponents. Therefore, no further discussion is required.

Input data set six contains the initial temperature guesses corresponding to the number of nodes receiving

secondary heat and is not used in this model because they have been set by the last entry in line five.

Input data set seven indicates the number of temperature dependent heat input curves and is not used in this model.

B. FEATURES

TMDL is presented to the user in discrete sections. Upon entry into the first section, the user is offered an optional overview. This option should be accepted on at least the first run. At the completion of the overview the user is prompted for a data file name and title. The data file name should be changed for each successive run because TMDL does not over write existing files.

The second section provides prompts for the physical characteristics of the structure. Initially, the user is given a list of four aspect ratios from which to choose;

1.) 10 by 24 nodes provides a 1:2.4 ratio with 720 nodes.
2.) 15 by 15 nodes provides a 1:1 ratio with 675 nodes.
3.) 12 by 20 nodes provides a 1:1.6 ratio with 720 nodes.
4.) 8 by 30 nodes provides a 1:3.75 ratio with 720 nodes.

It should be noted that up to 30 additional nodes may be added depending on the existence of the mounting ear. Figure 16 displays the specific geometry with definitions.

After selecting the desired aspect ratio, the user must specify the use of either SI or English units. The input of

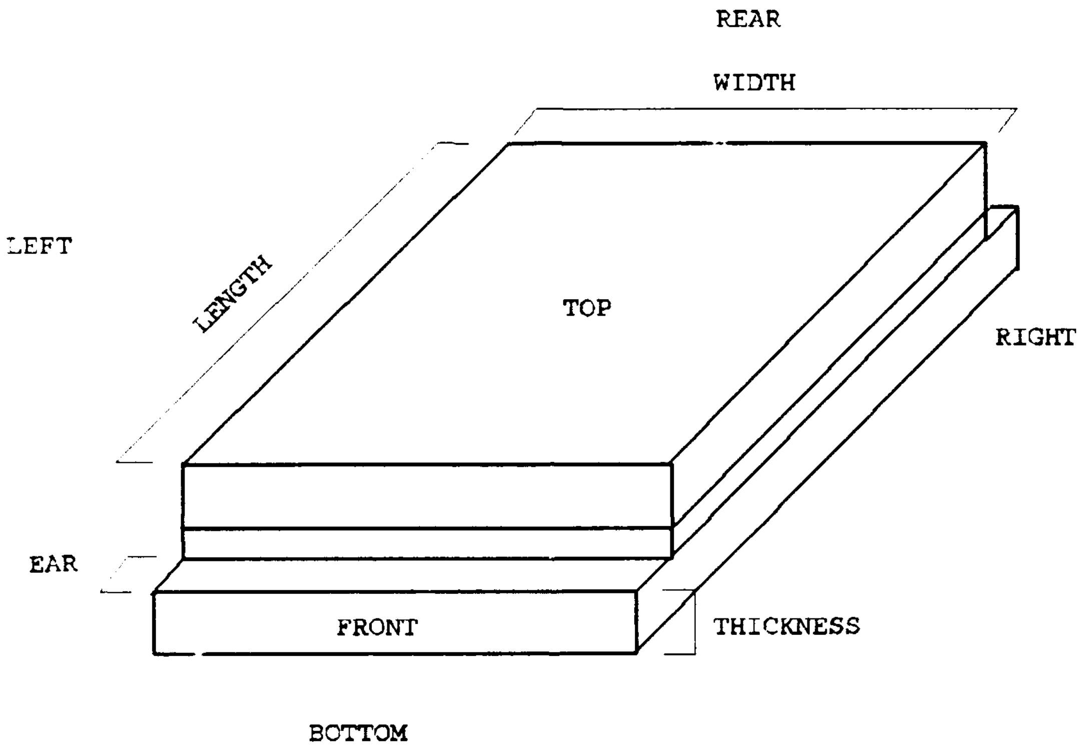

REAR
WIDTH
LEFT
LENGTH
TOP
RIGHT
EAR
FRONT
THICKNESS
BOTTOM

Figure 16.

data must remain consistent with this selection. At this point the input of structure characteristics are requested. These entries consist of the length, width, thickness, and thermal conductivity of each layer. There is ample opportunity for correction or alteration throughout TMDL.

Section three requests the input of initial and ambient temperatures. TMDL has six ambient temperatures.

The final section allows for the input of external heat sources. TMDL provides for heat input into the upper substrate surface only. There are four methods of external heat input from which to choose. The first alternative allows for a total rate of heat applied to the upper surface. An entry for this alternative would be divided by the nodes and distributed appropriately. Option two provides for the entry of average heat per unit area. The third alternative gives the user the freedom to enter heat in specifically selected nodes. Option four offers no heat input.

At the conclusion of these entries an output data file will be created in the proper format and placed in a file named by the user.

APPENDIX A

TMDL LISTING

```
$LARGE
C
C  TITLE:    MODEL BUILDER
C  AUTHOR:   LT PATRIC ROESCH
C  DATE:     15 MARCH 1991
C  COMPILER: MICROSOFT VERSION 4.01
C  LINKER:   MICROSOFT VERSION 3.55
C
C  DEFINE REAL VARIABLES
      REAL SL,SW,DELS,KS,EW,EL,DELE,KE,CL,CW,DELC,KC,IT,UPRT,LWRT,LT,RT
     +,FT,BT,THEAT,THPN,AHEAT,NHEAT,DELX,DELY,SYLR,SXFB,SXX,SYY,SZE,SZT
      REAL EXFB,EYY,EXX,EZC,EAR,CYLRE,CYYE,CXFBE,CXXE,CZEAR,CYLR,CYY,CXX
     +,CZB,ACC,DAMP,CONFAC,EYLR
C
C
C  DEFINE INTEGER VARIABLES
      INTEGER NPL,NWIDE,NDEEP,NUM,NUMA,CH,H,TOTNOD,NN,NC,N,IB,IE,IC,ID,
     +COUNT,CONTEMP,USEL,ZER,NMAX,TMAX,HTRS,D1,D2,D3,D5,D4,D6,D7,MAXIT
C
C  DEFINE ALL ONE CHARACTER VARIABLES
      CHARACTER*1 ANS,ANSN,ANSA,ANSSL,ANSSW,ANSST,ANSSK,ANSE,ANSEL,
     +ANSEW,ANSET,ANSEK,ANSC,ANSCL,ANSCW,ANSCT,ANSCK,ANST,ANSTI,ANSTU,
     +ANSTLR,ANSTL,ANSTR,ANSTF,ANSTB,ANSH,ATH,ANSHA,AHN,SELECT,SELH
     +,ANSWER,DELT
C
C  CHARACTER VARIABLES OF MORE THAN ONE POSITION
      CHARACTER DATAF*70,UK*11,UT*1,UH*13,UAH*6,NAME*6,UL*2
C
C  DEFINE MATRICES
      REAL HEAT(30,15),COEF(740,9)
C
      INTEGER IH(240),JH(240),NCON(740,9)
C
C
C
C  PROVIDE THE USER WITH A PROGRAM OVERVIEW
C
1000  CALL CLS
      WRITE(*,1001)
1001  FORMAT(///,'                                                  ',/,
     +'   THIS PROGRAM WAS WRITTEN TO INTEGRATE WITH EXISTING   ',/,
     +'   THERMAL ANALYSIS SOFTWARE AND TO REDUCE THE AMOUNT    ',/,
     +'   OF TIME REQUIRED FOR DATA ENTRY.',//,
     +'   WOULD YOU LIKE AN OVERVIEW OF THE PROGRAM PRIOR TO    ',/,
     +'   BEGINNING? ENTER Y FOR YES OR N FOR NO.  ',2X,\)
      READ(*,1002) OVR
1002  FORMAT(A1)
```

```
C
      IF (OVR.EQ.'Y'.OR.OVR.EQ.'y') THEN
        CALL CLS
        WRITE(*,1003)
1003    FORMAT(//,'              **********OVERVIEW*********     ',//,
     +'   THIS PROGRAM PERFORMS A NODAL ANALYSIS OF A THREE    ',/,
     +'   LAYERED CONDUCTIVE STRUCTURE (SUBSTRATE,EPOXY,AND   ',/,
     +'   CARRIER). THE OUTPUT CONSISTS OF UP TO 740 COEFFI-  ',/,
     +'   CIENTS THAT CONTRIBUTE TO THE DETERMINATION OF      ',/,
     +'   THE TEMPERATURE DISTRIBUTION OF A MICROCIRCUIT      ',/,
     +'   WHEN FED INTO THE THERMAL ANALYZER. ',/,
     +'   THE FOLLOWING IS AN ORDERED OUTLINE OF THE MAJOR     ',/,
     +'   SECTIONS OF THIS PROGRAM AND WHAT ENTRIES ARE    ',/,
     +'   REQUIRED OF THE USER.  ',/,
     +'   NOTE THAT ALL ENTRIES WILL BE IN UPPER CASE LETTERS.  ',/,
     +'   A.) DATA FILE  ',/,
     +'      1.) ENTER THE TITLE OF THE OUTPUT DATA FILE.  ',/,
     +'      2.) ENTER THE DATA FILE NAME. THIS PROGRAM WILL NOT',/,
     +'         ERASE OR WRITE OVER EXISTING FILES.  ',//,
     +'  PLEASE PRESS ENTER WHEN READY TO CONTINUE.  ',2X,\)
        READ(*,1002)ANSWER
C
        CALL CLS
        WRITE(*,1004)
1004    FORMAT(///,'            ',/,
     +'   B.) STRUCTURE PHYSICAL CHARACTERISTICS     ',/,
     +'      1.) SELECT UNIT TYPE (SI OR ENGLISH)    ',/,
     +'      2.) SELECT FROM FOUR ALTERNATIVES THE DESIRED NODAL',/,
     +'         ASPECT RATIO.  ',/,
     +'      3.) ENTER EACH LAYER CHARACTERISTICS. SUBSTRATE IS ',/,
     +'         THE TOP LAYER, FOLLOWED BY EPOXY AND THE CARRIER',/,
     +'         LAYERS, RESPECTIVELY. THE CARRIER LAYER MAY    ',/,
     +'         CONTAIN AN EAR ON THE FRONT AND BACK SURFACES ',/,
     +'         IF SPECIFICATIONS REQUIRE IT.',//,
     +'   PLEASE PRESS ENTER WHEN READY TO CONTINUE.  ',2X,\)
        READ(*,1002)ANSWER
C
        CALL CLS
        WRITE(*,1005)
1005    FORMAT(///,
     +'  C.) INITIAL AND AMBIENT TEMPERATURES      ',/,
     +'     1.) ENTER INITIAL CHIP AND CHASIS TEMPERATURE.  ',/,
     +'        ASSUME STEADY STATE FOR INITIAL TEMPERATURE.',/,
     +'     2.) ENTER AMBIENT TEMPERATURES FROM ALL SIDES OF THE',/,
     +'        STRUCTURE.',//,
     +'  D.) HEAT INPUT  ',/,
     +'     HEAT INJECTION OCCURS ONLY ON THE UPPER SURFACE OF ',/,
     +'     THE SUBSTRATE LAYER. THIS PROGRAM SUPPLIES THE USER ',/,
     +'     FOR ALTERNATIVE METHODS FOR ENTERING HEAT;  ',/,
     +'     1.) TOTAL HEAT OVER SUFACE   ',/,
     +'     2.) AVERAGE HEAT PER UNIT AREA  ',/,
     +'     3.) INPUT HEAT NODE BY NODE  ',/,
     +'     4.) NO HEAT INPUT   ',//,
     +'  THIS COMPLETES THE PROGRAM OVERVIEW. PLEASE PRESS ENTER ',/,
     +'  TO CONTINUE.  ',2X,\)
```

```
      READ(*,1002)ANSWER
C
     ELSEIF (OVR.EQ.'N'.OR.OVR.EQ.'n') THEN
      GOTO 1007
     ELSE
      GOTO 1006
1007 ENDIF
C
C
C
C  INITIALIZE MATRICES
C
     DATA HEAT /450*0.0/
     DATA IH /240*0/
     DATA JH /240*0/
C
C  VARIABLE, CONSTANT, AND STRING DEFINITION
C     PHYSICAL CHARACTERISTICS
C       SL,EL,CL - SUBSTRATE,EPOXY, AND CARRIER LENGTHS
C       SW,EW,CW - SUBSTRATE,EPOXY, AND CARRIER WIDTHS
C       KS,KE,KC - SUBSRATE,EPOXY,AND CARRIER THERMAL CONDUCTIVITIES
C       DELS,DELE,DELC - SUBSTRATE,EPOXY AND CARRIER THICKNESS
C       DELX - SL/NDEEP
C       DELY - SW/NWIDE
C       NPL - NUMBER OF NODES PER A LAYER
C       NWIDE - NUMBER OF NODES WIDE
C       NDEEP - NUMBER OF NODES DEEP
C       UL - UNITS OF LENGTH (SI OR ENGLISH)
C       UK - UNITS OF THERMAL CONDUCTIVITY (SI OR ENGLISH)
C       EAR - DEPTH OF THE EAR (SL-CL)/2
C
C     INITIAL AND AMBIENT TEMPERATURES
C       IT - INITIAL CHASIS AND CHIP TEMPERATURE
C       LT - LEFT SIDE AMBIENT TEMPERATURE
C       RT - RIGHT SIDE AMBIENT TEMPERATURE
C       FT - FRONT AMBIENT TEMPERATURE
C       BT - BACK AMBIENT TEMPERATURE
C       UPRT - UPPER AMBIENT TEMPERATURE
C       LWRT - LOWER AMBIENT TEMPERATURE
C       UT - UNITS OF TEMPERATURE (CENTIGRADE OR FARENHIET)
C
C     HEAT INPUT
C       THEAT - TOTAL INJECTED HEAT
C       THPN - TOTAL HEAT PER NODE
C       AHEAT - AVERAGE HEAT OVER A GIVEN SURFACE
C       NHEAT - HEAT PER NODE INJECTED NODE BY NODE
C       UH AND UAH - UNITS OF HEAT (SI OR ENGLISH)
C       NUM,NUMA,CH,H - DUMMY VARIABLES USED TO CREATE THE VECTORS
C                  IH, & JH WHICH ARE USED TO RELATE NODE
C                  NUMBER TO MATRIX POSITION
C       TOTNOD,NN,NC,N - VARIABLES USED TO ALLOW HEAT INPUT NODALLY
C       IH,JH - VECTORS USED TO CORRELATE NODE NUMBER WITH
C             MATRIX POSITION
C       HEAT - MATRIX USED TO CONTAIN HEAT INPUTS
C
```

```
C     COEFFICIENT DEFINITIONS   Y IMPLIES WIDTH & X IMPLIES DEPTH
C                      Z IMPLIES HIEGHT
C                      S IMPLIES SUBSTRATE, E IMPLIES EPOXY
C                      C IMPLIES CARRIER
C
C        SYLR - PROVIDES COEFFICIENT FOR  LEFT OR RIGHT EDGE NODES
C           TO THE EXTERNAL NODE. SYLR IMPLIES SUBSTRATE,
C           EYLR - EPOXY
C           CYLR - CARRIER/ NO EAR
C           CYLRE - CARRIER W/ EAR
C
C        SXFB - PROVIDES COEFFICIENT FOR FRONT AND BACK EDGE NODES
C           TO THE EXTERNAL NODE.
C           EXFB - EPOXY
C           CXFB - CARRIER/ NO EAR
C           CXFBE - CARRIER W/ EAR
C
C        SYY - PROVIDES INTERNAL COEFFICIENT IN THE 'Y' DIRECTION
C           EYY - EPOXY
C           CYY - CARRIER W/NO EAR
C           CYYE - CARRIER W/ EAR
C
C        SXX - PROVIDE INTERNAL COEFFICIENT IN THE 'X' DIRECTION
C           EXX - EPOXY
C           CXX - CARRIER W/NO EAR
C           CXXE - CARRIER W/ EAR
C
C        SZT - COEFFICIENT FOR SUBSTRATE TO UPPER EXTERNAL NODE
C        SZE - COEFFICIENT FOR SUBSTRATE TO EPOXY
C        EZC - COEFFICIENT FOR EPOXY TO CARRIER
C        CZEAR - COEFFICIENT FOR EAR TO LOWER EXTERNAL NODE
C        CZB - COEFFICIENT FOR CARRIER TO LOWER EXTERNAL NODE
C
C     CHARACTER
C       ALL ONE CHARACTER STRINGS EXCEPT 'UT' ARE SIMPLE
C       YES, NO, OR SELECTION ANSWERS AND DO NOT REQUIRE
C       EXPLANATION.
C
C     DATA FILE
C       DATAF - TITLE FOR DATA FILE, LINE ONE
C       NAME - DATA FILE NAME
C       IB,IE,ID,IC - COUNTERS FOR FILLING DATA FILE
C       COUNT - TOTAL NUMBER OF NODES
C       ACC - MINIMUM ERROR CRITERIA AT WHICH CALCULATIONS CEASE
C       DAMP - DAMPING FACTOR FOR PREVENTION OF TEMPERATURE
C           OSCILLATIONS
C       CONFAC - CONVERGENCE FACTOR USED TO INCREASE DAMPING
C       CONTEMP - TMDL HAS 6 CONSTANT AMBIENT TEMPERATURES
C       USEL - TELLS ANALYZER WHAT TYPE OF UNITS ARE IN USE
C       ZER - DUMMY VARIABLE
C       NMAX - MAXIMUM NODES PROGRAM IS CAPABLE OF EVALUATING
C       TMAX - MAXIMUM NUMBER OF CONSTANT TEMPERATURES PROGRAM
C           CAN ACCEPT
C       HTRS - NOT APPLICABLE TO THIS PROGRAM - NO EXPLANATION
C           NECCESSARY
```

```
C           D1-7 - DATA SETS IN USE
C           MAXIT - MAXIMUM ITERATIONS COMPUTER WILL PERFORM TO
C                ACHIEVE SOLUTION
C
C   BEGIN PROGRAM
C
      CALL CLS
      WRITE(*,406)
406   FORMAT( ,'         **********************************************
     +*****',
     +'         ************************************************',
     +'         **                                         **',/,
     +'         **      PLEASE SELECT UPPER CASE LETTERS       **',
     +'         **      PRIOR TO BEGINNING.               **',
     +'         **                                         **',
     +'         ************************************************',
     +'         ************************************************')
C
      WRITE(*,201)
201   format( ,'                                            ',/,
     +'         PRIOR TO ENTERING DATA INTO THIS PROGRAM ENSURE THAT',/,
     +'         YOU HAVE A DRAWING OF YOUR DESIGN AND ALL PERTINENT ',/,
     +'         DATA.                                  ',//,
     +'         PRESS ENTER WHEN READY TO CONTINUE.',2X,\)
      READ(*,304)ANSWER
C
C   ALLOW USER TO NAME THE DATA FILE
      CALL CLS
C
      WRITE(*,701)
701   FORMAT( ,'     THIS PROGRAM CREATES AN OUTPUT DATA FILE FOR ENTRY
     + INTO THE',/,
     +'     EXISTING THERMAL ANALYZER. FURTHERMORE, THIS PROGRAM DOES',/,
     +'     NOT ERASE OR WRITE OVER THE EXISTING DATA FILE. THEREFORE',/,
     +'     THE USER WILL NAME THE DATA FILE FOR EACH RUN OF THIS    ',/,
     +'     PROGRAM. THE FILE NAME IS LIMITED TO SIX CHARACTERS.    ',/,
     +'     PLEASE ENTER THE DESIRED DATA FILE NAME.  ',2X,\)
      READ 702, NAME
702   FORMAT(A6)
C
C
C
C   ALLOW USER TO PROVIDE TITLE LINE FOR DATA FILE
      CALL CLS
      WRITE(*,202)
202   FORMAT(////,'        ENTER THE DESIRED TITLE TO BE PLACED ON LINE
     +',/,'        NUMBER ONE OF THE OUTPUT DATA FILE.',/,
     +'          ',2X,\)
      READ 1,DATAF
1     FORMAT(A70)
C
C     CALL CLS
C
C
C   SUPPLY A LISTING OF ACCEPTABLE ASPECT RATIOS
```

```
C
      WRITE(*,203)
203   FORMAT(//,'     THE FOLLOWING ASPECT RATIOS ARE AVAILABLE.')
      WRITE(*,204)
204   FORMAT(/,'     1.) 10 BY 24 PROVIDES A 1:2.4 RATIO WITH 740 NODES.
     +',/,'     2.) 15 BY 15 PROVIDES A 1:1 RATIO WITH 675 NODES.',/,
     +'     3.) 12 BY 20 PROVIDES A 1:1.6 RATIO WITH 720 NODES.',/
     +,'     4.) 8 BY 30 PROVIDES A 1:3.75 RATIO WITH 720 NODES.',//
     +,'     PLEASE SELECT A NUMBER ONE THROUGH FOUR.  ',2X,\)
      READ(*,3020)SELECT
3020  FORMAT(A1)
C
C   USER CAN CONTINUE OR MAKE ANOTHER SELECTION
      CALL CLS
      IF (SELECT.NE.'1'.AND.SELECT.NE.'2'.AND.SELECT.NE.'3'.AND.SELECT.
     +NE.'4') THEN
        GOTO 7
      ELSE
10      WRITE(*,3) SELECT
3       FORMAT(  ,'          YOU SELECTED NUMBER ',A1,' OF THE FOLLO
     +WING 4 ALTERNATIVES.',,
     +'          1.) 10 BY 24 FOR A 1:2.4 RATIO',/,
     +'          2.) 15 BY 15 FOR A 1:1 RATIO',/,
     +'          3.) 12 BY 20 FOR A 1:1.6 RATIO',,
     +'          4.) 8 BY 30 FOR A 1:3.75 RATIO',/)
C
      ENDIF
C
5     WRITE(*,205)
205   FORMAT('         IS THIS THE DESIRED SELECTION. ENTER Y FOR YES AN
     +D N FOR NO.  ',2X,\)
      READ(*, 8)ANS
8     FORMAT(A1)
C
C
      IF (ANS.EQ.'N') THEN
        GOTO 7
      ELSEIF (ANS.EQ.'Y') THEN
        GOTO 9
      ELSE
        CALL CLS
        GOTO 10
9     ENDIF
C
C
      CALL CLS
C
C
C   AFTER CONFIRMATION OF SELECTION FILL CONSTANTS WITH APPROPRIATE
C   VALUES
      IF (SELECT.EQ.'1') THEN
        NWIDE = 10
        NDEEP = 24
        NPL = 240
C
```

```
      ELSEIF (SELECT.EQ.'2') THEN
        NWIDE = 15
        NDEEP = 15
        NPL = 225
C
      ELSEIF (SELECT.EQ.'3') THEN
        NWIDE = 12
        NDEEP = 20
        NPL = 240
C
      ELSEIF (SELECT.EQ.'4') THEN
        NWIDE = 8
        NDEEP = 30
        NPL = 240
      ENDIF
C
C   MAKE DESIRED UNIT SELECTION
C
2     WRITE(*,206)
206   FORMAT(//////,'        THIS PROGRAM IS CAPABLE OF OPERATIONS IN EI
     +THER SI OR ',/,
     +'        ENGLISH UNITS. AFTER THE SELECTION OF THE UNITS, ALL ',/,
     +'        ENTRIES MUST BE COMPATIBLE. PLEASE MAKE YOUR SELECTION.',/
     +,'        S FOR SI NOTATION',/,
     +'        E FOR ENGLISH NOTATION  ',2X,$)
      READ(*,302)ANSN
302   FORMAT(A1)
C
C   CHECK FOR CORRECT UNIT SELECTION
4     IF (ANSN.EQ.'S') THEN
        WRITE(*,207)
207     FORMAT(///,'        YOU SELECTED SI NOTATION.')
      ELSEIF (ANSN.EQ.'E') THEN
        WRITE(*,208)
208     FORMAT(///,'        YOU HAVE SELECTED ENGLISH NOTATION.')
      ELSE
        CALL CLS
        GOTO 2
      ENDIF
C
      WRITE(*,209)
209   FORMAT('        IS THIS THE DESIRED SELECTION? ENTER Y FOR YES AND
     +N FOR NO.  ',2X,$)
      READ(*,303)ANSA
303   FORMAT(A1)
C
C
C   SHOULD I CONTINUE OR RESELECT UNITS
      IF (ANSA.EQ.'Y') THEN
        GOTO 6
      ELSEIF (ANSA.EQ.'N') THEN
        CALL CLS
        GOTO 2
      ELSE
        CALL CLS
```

```
      WRITE(*,210)
210      FORMAT(//////,'       THIS PROGRAM IS CAPABLE OF OPERATIONS IN
     + EITHER SI OR',/,
     +'       ENGLISH UNITS. AFTER THE SELECTION OF UNITS, ALL',/,
     +'       ENTRIES MUST BE COMPATIBLE. PLEASE MAKE YOUR SELECTION.',/
     +,'       S FOR SI NOTATION.',/,
     +'       E FOR ENGLISH NOTATION.')
C
      GOTO 4
6     ENDIF
C
      CALL CLS
C
C*****************************************************************************
C***************** ENTER CHIP CHARACTERISTICS FOR EACH LAYER***********
C*****************************************************************************
C
C
C****************************SUBSTRATE LAYER**************************
C
      WRITE(*,211)
211   FORMAT(///,'*******************************************************
     +*******************',/,
     +'*************************SUBSTRATE CHARACTERISTICS***************
     +*********',/,
     +'****************************************************************
     +*********',/)
C
C   PROVIDE CORRECT UNIT ABBREVIATIONS
      IF (ANSN.EQ.'S') THEN
        WRITE(*,212)
212     FORMAT(' ALL ENTRIES ARE IN SI NOTATION.',/)
        UL = 'cm'
        UK = 'Watts/cm/C '
        UT = 'C'
      ELSEIF (ANSN.EQ.'E') THEN
        WRITE(*,213)
213     FORMAT(' ALL ENTRIES ARE IN ENGLISH NOTATION.',/)
        UL = 'in'
        UK = 'Btu/hr/in/F'
        UT = 'F'
      ENDIF
C
C
C
      WRITE(*,220)UL
220   FORMAT(' ENTER SUBSTRATE LENGTH (',A2,'):          ',2X,\)
      READ *,SL
C
      WRITE(*,216)UL
216   FORMAT(/,' ENTER SUBSTRATE WIDTH (',A2,'):           ',2X,\)
      READ *,SW
C
      WRITE(*,217)UL
217   FORMAT(/,' ENTER SUBSTRATE THICKNESS (',A2,'):       ',2X,\)
```

```
      READ *,DELS
C
      WRITE(*,218)UK
218   FORMAT(/,' ENTER SUBSTRATE THERMAL CONDUCTIVITY (',A12,'): ',2X,\)
      READ *,KS
C
31    CALL CLS
C
C  MAKE ANY CHANGES OR CORRECTIONS TO SUBSRATE ENTRIES
C
      WRITE(*,214)
214   FORMAT(//,'        YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE SUB
     +STRATE:',//)
20    PRINT *,'      1.) LENGTH     ',SL
      PRINT *,'      2.) WIDTH      ',SW
      PRINT *,'      3.) THICKNESS  ',DELS
      PRINT *,'      4.) k          ',KS
C
      WRITE(*,215)
215   FORMAT(/,'      DO YOU WISH TO MAKE ANY CHANGES? SELECT Y FOR YES
     + AND N FOR NO.  ',2X,\)
      READ(*,304)ANSS
304   FORMAT(A1)
C
      IF (ANSS.EQ.'Y') THEN
12      CALL CLS
        WRITE(*,219)
219     FORMAT(////)
        PRINT *,'    THE CURRENT ENTRY FOR LENGTH IS ',SL,' ',UL
        WRITE(*,2190)
2190    FORMAT(/,'    WOULD YOU LIKE TO CHANGE THE LENGTH? (Y OR N)
     +  ',2X,\)
        READ(*,304)ANSSL
        PRINT *
          IF (ANSSL.EQ.'Y') THEN
            WRITE(*,221)UL
221         FORMAT(/,'    ENTER SUBSTRATE LENGTH (',A2,'):  ',2X,\)
            READ *,SL
          ELSEIF (ANSSL.EQ.'N') THEN
            GOTO 14
          ELSE
            GOTO 12
          ENDIF
C
14      CALL CLS
        WRITE(*,2191)
2191    FORMAT(////)
        PRINT *,'    THE CURRENT ENTRY FOR WIDTH IS ',SW,' ',UL
        WRITE(*,222)
222     FORMAT(/,'    WOULD YOU LIKE TO CHANGE THE WIDTH? (Y OR N)  ',
     +2X,\)
        READ(*,2192) ANSSW
2192    FORMAT(A1)
          IF (ANSSW.EQ.'Y') THEN
            WRITE(*,223)UL
```

```
223         FORMAT(/,'     ENTER SUBSTRATE WIDTH (',A2,'):  ',2X,\)
            READ *,SW
          ELSEIF (ANSSW.EQ.'N') THEN
            GOTO 13
          ELSE
            GOTO 14
          ENDIF
C
C
13      CALL CLS
        WRITE(*,2193)
2193    FORMAT(////)
        PRINT *,'     THE CURRENT ENTRY FOR THICKNESS IS ',DELS,' ',UL
        WRITE(*,224)
224     FORMAT(/,'     WOULD YOU LIKE TO CHANGE THE THICKNESS? (Y OR N)
     +  ',2X,\)
        READ(*,2194) ANSST
2194    FORMAT(A1)
          IF (ANSST.EQ.'Y') THEN
            WRITE(*,225)UL
225         FORMAT(/,'     ENTER SUBSTRATE THICKNESS (',A2,'):  ',2X,
     +\)
            READ *,DELS
          ELSEIF (ANSST.EQ.'N') THEN
            GOTO 15
          ELSE
            GOTO 13
          ENDIF
C
C
15      CALL CLS
        WRITE(*,2195)
2195    FORMAT(////)
        PRINT *,'     THE CURRENT ENTRY FOR THERMAL CONDUCTIVITY IS ',KS
     +,' ',UK
        WRITE(*,226)
226     FORMAT(/,'     WOULD YOU LIKE TO CHANGE THE THERMAL CONDUCTIVIT
     +Y? (Y OR N)  ',2X,\)
        READ(*,2196) ANSSK
2196    FORMAT(A1)
          IF (ANSSK.EQ.'Y') THEN
            WRITE(*,227)UK
227         FORMAT(/,'     ENTER SUBSTRATE THERMAL CONDUCTIVITY (',A1
     +2,'):  ',2X,\)
            READ *,KS
          ELSEIF (ANSSK.EQ.'N') THEN
            GOTO 16
          ELSE
            GOTO 15
          ENDIF
C
C  ALLOW ANOTHER REVIEW OF SUBSTRATE DATA ENTRIES
C
16      CALL CLS
        WRITE(*,228)
```

```
228    FORMAT(///,'      YOU HAVE MADE THE FOLLOWING CORRECTIONS TO T
     +HE SUBSTRATE ENTRIES.'//)
       GOTO 20
C
C
      ELSEIF (ANSS.EQ.'N') THEN
        GOTO 17
      ELSE
        GOTO 31
      ENDIF
C
C
C*****************************EPOXY CHARACTERISTICS******************
C
C
17    CALL CLS
      WRITE(*,229)
229   FORMAT(///,'*******************************************************
     +******************'/,
     +'***************************EPOXY CHARACTERISTICS*****************
     +*******'/,
     +'*****************************************************************
     +*******'//,
     +'   NOTE THAT SUBSTRATE AND EPOXY LENGTH AND WIDTH ARE EQUAL. '/
     +)
C
      WRITE(*,230)UL
230   FORMAT(/,'   ENTER EPOXY THICKNESS (',A2,'): ',2X,\)
      READ *,DELE
      WRITE(*,231)UK
231   FORMAT(/,'   ENTER EPOXY THERMAL CONDUCTIVITY (',A12,'): ',2X,\)
      READ *,KE
C
      EL = SL
      EW = SW
C
C  REVIEW EPOXY ENTRIES
C
30    CALL CLS
      WRITE(*,232)
232   FORMAT(///,'    YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE EPOXY
     + LAYER.'/)
32    PRINT *,'   1.) LENGTH                 ',EL
      PRINT *,'   2.) WIDTH                  ',EW
      PRINT *,'   3.) THICKNESS              ',DELE
      PRINT *,'   4.) k                      ',KE
C
      WRITE(*,233)
233   FORMAT(/,'    NOTE THAT CHANGING EPOXY LENGTH OR WIDTH ALSO CHANG
     +ES',/,'    SUBSTRATE LENGTH OR WIDTH.'//,
     +'    DO YOU WISH TO MAKE ANY CHANGES TO THE EPOXY ENTRIES? (Y OR
     +N)  ',2X,\)
      READ(*,2197) ANSE
2197  FORMAT(A1)
C
```

```
C   MAKE ANY CHANGES OR CORRECTIONS TO EPOXY ENTRIES
C
     IF (ANSE.EQ.'Y') THEN
22        CALL CLS
          WRITE(*,2198)
2198      FORMAT(////)
          PRINT *,'    THE CURRENT ENTRY FOR LENGTH IS ',EL,' ',UL
          WRITE(*,234)
234       FORMAT(/,'    WOULD YOU LIKE TO CHANGE THE LENGTH? (Y OR N)
     +',2X,\)
          READ(*,2199) ANSEL
2199      FORMAT(A1)
C
            IF (ANSEL.EQ.'Y') THEN
              WRITE(*,235)UL
235            FORMAT(/,'    ENTER EPOXY LENGTH (',A2,'):  ',2X,\)
              READ *,EL
              SL = EL
            ELSEIF (ANSEL.EQ.'N') THEN
              GOTO 21
            ELSE
              GOTO 22
21           ENDIF
C
24        CALL CLS
          WRITE(*,601)
601       FORMAT(////)
          PRINT *,'    THE CURRENT ENTRY FOR WIDTH IS ',EW,' ',UL
          WRITE(*,236)
236       FORMAT(/,'    WOULD YOU LIKE TO CHANGE THE WIDTH? (Y OR N)
     +',2X,\)
          READ(*,602)ANSEW
602       FORMAT(A1)
C
            IF (ANSEW.EQ.'Y') THEN
              WRITE(*,237)UL
237            FORMAT(/,'    ENTER EPOXY WIDTH (',A2,'):  ',2X,\)
              READ *,EW
              SW = EW
            ELSEIF (ANSEW.EQ.'N') THEN
              GOTO 23
            ELSE
              GOTO 24
23           ENDIF
C
26        CALL CLS
          WRITE(*,603)
603       FORMAT(////)
          PRINT *,'    THE CURRENT ENTRY FOR THICKNESS IS ',DELE,' ',UL
          WRITE(*,238)
238       FORMAT(/,'    WOULD YOU LIKE TO CHANGE THE THICKNESS? (Y OR N)
     + ',2X,\)
          READ(*,604)ANSET
604       FORMAT(A1)
C
```

```
        IF (ANSET.EQ.'Y') THEN
          WRITE(*,239)UT
239       FORMAT(,'     ENTER EPOXY THICKNESS (',A2,'):  ',2X,\)
          READ *,DELE
        ELSEIF (ANSET.EQ.'N') THEN
          GOTO 25
        ELSE
          GOTO 26
25      ENDIF
C
28    CALL CLS
      WRITE(*,605)
605   FORMAT(///)
      PRINT *,'    THE CURRENT ENTRY FOR THERMAL CONDUCTIVITY IS ',KE
   +,' ',UK
      WRITE(*,240)
240   FORMAT(/,'     WOULD YOU LIKE TO CHANGE THE THERMAL CONDUCTIVIT
   +Y? (Y OR N)  ',2X,\)
      READ(*,606)ANSEK
606   FORMAT(A1)
C
        IF (ANSEK.EQ.'Y') THEN
          WRITE(*,241)UK
241       FORMAT(,'     ENTER EPOXY THERMAL CONDUCTIVITY (',A12,')
   +:  ',2X,\)
          READ *,KE
        ELSEIF (ANSEK.EQ.'N') THEN
          GOTO 27
        ELSE
          GOTO 28
        ENDIF
C
C  PROVIDE ANOTHER REVIEW OF EPOXY ENTRIES
C
27    CALL CLS
      WRITE(*,242)
242   FORMAT(///,'     YOU HAVE MADE THE FOLLOWING CORRECTIONS TO THE EP
   +OXY ENTRIES.'//)
      GOTO 32
C
      ELSEIF (ANSE.EQ.'N') THEN
        GOTO 29
      ELSE
        GOTO 30
      ENDIF
C
C
C*****************************************CARRIER CHARACTERISTICS****************
C
C
29    CALL CLS
      WRITE(*,243)
243   FORMAT(//,'****************************************************
   +*********************'//,
   +'*****************************CARRIER CHARACTERISTICS*************
```

```
     +*********',/,
     +'***************************************************************
     +*********',)
C
944  WRITE(*,244)UL
244  FORMAT('  ENTER CARRIER LENGTH (',A2,'):    ',2X,\)
     READ *,CL
C
C  CARRIER LENGTH MUST BE GREATER THAN OR EQUAL TO SUBSTRATE LENGTH
C
     IF (CL.LT.SL) THEN
        CALL CLS
        WRITE(*,243)
        PRINT *,' CARRIER LENGTH MUST BE GREATER THAN OR EQUAL TO SUBS
     +TRATE LENGTH.  '
        GOTO 944
     ENDIF
C
C
     WRITE(*,246)UL
246  FORMAT(/,'  ENTER CARRIER THICKNESS (',A2,'):  ',2X,\)
     READ *,DELC
C
     WRITE(*,247)UK
247  FORMAT(/,'  ENTER CARRIER THERMAL CONDUCTIVITY (',A12,'): ',2X,\)
     READ *,KC
C
38   CALL CLS
     CW = SW
C
C  REVIEW CARRIER ENTRIES
C
     WRITE(*,248)
248  FORMAT(//,'    YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE CARRI
     +ER.',/)
36   PRINT *,'   1.) LENGTH          ',CL
     PRINT *,'   2.) WIDTH           ',CW
     PRINT *,'   3.) THICKNESS       ',DELC
     PRINT *,'   4.) k               ',KC
     PRINT *
     PRINT *,'   CHANGING WIDTH WILL ALSO CHANGE SUBSTRATE AND EPOXY W
     +IDTHS.'
C
     WRITE(*,249)
249  FORMAT(/,'    DO YOU WISH TO MAKE ANY CHANGES? (Y OR N)  ',2X,\)
     READ(*,607)ANSC
607  FORMAT(A1)
C
C  MAKE CORRECTIONS OR CHANGES TO CARRIER ENTRIES
C
     IF (ANSC.EQ.'Y') THEN
33      CALL CLS
        WRITE(*,608)
608     FORMAT(//)
933     PRINT *,'   THE CURRENT ENTRY FOR LENGTH IS ',CL,' ',UL
```

```
      WRITE(*,250)
250   FORMAT(/,'    WOULD YOU LIKE TO CHANGE LENGTH? (Y OR N)  ',2X,
     +\)
      READ(*,609)ANSCL
609   FORMAT(A1)
C
        IF (ANSCL.EQ.'Y') THEN
          WRITE(*,251)UL
251       FORMAT(/,'    ENTER CARRIER LENGTH (',A2,'):  ',2X,\)
          READ *,CL
          IF (CL.LT.SL) THEN
            CALL CLS
            WRITE(*,610)
610         FORMAT(////)
            PRINT *,'    CARRIER LENGTH MUST BE GREATER THAN OR EQ
     +UAL TO SUBSTRATE LENGTH.'
            PRINT *,'    THE PRESENT ENTRY FOR SUBSTRATE LENGTH IS
     + ',SL,' ',UL
            PRINT *
            GOTO 933
          ENDIF
C
        ELSEIF (ANSCL.EQ.'N') THEN
          IF (CL.LT.SL) THEN
            CALL CLS
            WRITE(*,611)
611         FORMAT(////)
            PRINT *,'    CARRIER LENGTH MUST BE GREATER THAN OR EQ
     +UAL TO SUBSTRATE LENGTH. '
            PRINT *,'    THE PRESENT ENTRY FOR SUBSTRATE LENGTH IS
     + ',SL,UL
            PRINT *
            GOTO 933
          ENDIF
          GOTO 39
        ELSE
          GOTO 33
        ENDIF
C
39    CALL CLS
      WRITE(*,612)
612   FORMAT(///)
      PRINT *,'    THE CURRENT ENTRY FOR WIDTH IS ',CW,' ',UL
      PRINT *,'    CHANGING THIS ENTRY WILL CHANGE SUBSTRATE AND '
      PRINT *,'    EPOXY WIDTHS. THEY ARE ALL EQUAL. '
      WRITE(*,252)
252   FORMAT(/,'    WOULD YOU LIKE TO CHANGE WIDTH? (Y OR N)  ',2X,\
     +)
      READ(*,613)ANSCW
613   FORMAT(A1)
C
        IF (ANSCW.EQ.'Y') THEN
          WRITE(*,253)UL
253       FORMAT(/,'    ENTER CARRIER WIDTH (',A2,'):  ',2X,\)
          READ *,CW
```

```
          SW = CW
          EW = CW
        ELSEIF (ANSCW.EQ.'N') THEN
          GOTO 34
        ELSE
          GOTO 39
        ENDIF
C
34      CALL CLS
        WRITE(*,614)
614     FORMAT(////)
        PRINT *,'    THE CURRENT ENTRY FOR THICKNESS ',DELC,' ',UL
        WRITE(*,254)
254     FORMAT(/,'     WOULD YOU LIKE TO CHANGE THICKNESS? (Y OR N)  ',
     +2X,\)
        READ(*,615)ANSCT
615     FORMAT(A1)
C
        IF (ANSCT.EQ.'Y') THEN
          WRITE(*,2551)UL
2551        FORMAT(/,'     ENTER CARRIER THICKNESS (',A2,'):  ',2X,\)
          READ *,DELC
        ELSEIF (ANSCT.EQ.'N') THEN
          GOTO 35
        ELSE
          GOTO 34
        ENDIF
C
35      CALL CLS
        WRITE(*,616)
616     FORMAT(////)
        PRINT *,'    THE CURRENT ENTRY FOR THERMAL CONDUCTIVITY IS ',KC
     +,' ',UK
        WRITE(*,256)
256     FORMAT(/,'     WOULD YOU LIKE TO CHANGE THE THERMAL CONDUCTIVIT
     +Y? (Y OR N)  ',2X,\)
        READ(*,617)ANSCK
617     FORMAT(A1)
C
        IF (ANSCK.EQ.'Y') THEN
          WRITE(*,257)UK
257         FORMAT(/,'     ENTER CARRIER THERMAL CONDUCTIVITY (',A12,
     +'):  ',2X,\)
          READ *,KC
        ELSEIF (ANSCK.EQ.'N') THEN
          GOTO 40
        ELSE
          GOTO 35
        ENDIF
C
C  ALLOW FOR ANOTHER REVIEW OF CARRIER ENTRIES
C
40    CALL CLS
      WRITE(*,258)
258   FORMAT(///,'     YOU HAVE MADE THE FOLLOWING CORRECTIONS TO THE C
```

```
     +ARRIER ENTRIES.'//)
      GOTO 36
C
C
      ELSEIF (ANSC.EQ.'N') THEN
        GOTO 37
      ELSE
        GOTO 38
37    ENDIF
C
C
C
C************************************************************************
C******************TEMPERATURE INPUTS**********************************
C************************************************************************
C
      CALL CLS
      WRITE(*,260)
260   FORMAT(//,'*******************************************************
     +******************'/,
     +'***************************AMBIENT TEMPERATURE INPUT************
     +*********'/,
     +'***************************************************************
     +*********'//)
C
C   INITIAL TEMPERATURE
C
      WRITE(*,261)UT
261   FORMAT('     ENTER THE INITIAL CHIP TEMPERATURE (',A1,'). ',2X,\)
      READ *,IT
C
C   UPPER AMBIENT TEMPERATURE
C
      WRITE(*,262)UT
262   FORMAT(/,'     ENTER THE UPPER SURFACE AMBIENT TEMPERATURE (',A1,'
     +). ',2X,\)
      READ *,UPRT
C
C   LOWER AMBIENT TEMPERATURE
C
      WRITE(*,263)UT
263   FORMAT(/,'     ENTER THE LOWER SURFACE AMBIENT TEMPERATURE (',A1,'
     +). ',2X,\)
      READ *,LWRT
C
C   RIGHT SIDE AMBIENT TEMPERATURE
C
      WRITE(*,264)UT
264   FORMAT(/,'     ENTER THE RIGHT SURFACE AMBIENT TEMPERATURE (',A1,'
     +). ',2X,\)
      READ *,RT
C
C   LEFT SIDE AMBIENT TEMPERATURE
C
      WRITE(*,265)UT
```

```
265  FORMAT(/,'     ENTER THE LEFT SURFACE AMBIENT TEMPERATURE (',A1,')
    +. ',2X,\)
     READ *,LT
C
C   FRONT AMBIENT TEMPERATURE
C
     WRITE(*,266)UT
266  FORMAT(/,'     ENTER THE FRONT SURFACE AMBIENT TEMPFRATURE (',A1,'
    +). ',2X,\)
     READ *,FT
C
C   REAR SIDE AMBIENT TEMPERATURE
C
     WRITE(*,267)UT
267  FORMAT(/,'     ENTER THE REAR SURFACE AMBIENT TEMPERATURE (',A1,')
    +. ',2X,\)
     READ *,BT
C
C
C   REVIEW TEMPERATURE ENTRIES
C
50   CALL CLS
     WRITE(*,268)
268  FORMAT(////,'     YOU HAVE MADE THE FOLLOWING AMBIENT TEMPERATURE
    +ENTRIES.',/)
     PRINT *,'  1.) INITIAL CHIP TEMPERATURE        ',IT,' ',UT
     PRINT *,'  2.) UPPER AMBIENT TEMPERATURE        ',UPRT,' ',UT
     PRINT *,'  3.) LOWER AMBIENT TEMPERATURE        ',LWRT,' ',UT
     PRINT *,'  4.) RIGHT AMBIENT TEMPERATURE        ',RT,' ',UT
     PRINT *,'  5.) LEFT AMBIENT TEMPERATURE        ',LT,' ',UT
     PRINT *,'  6.) FRONT AMBIENT TEMPERATURE        ',FT,' ',UT
     PRINT *,'  7.) REAR AMBIENT TEMPERATURE        ',BT,' ',UT
C
C
     WRITE(*,269)
269  FORMAT(/,'     WOULD YOU LIKE TO MAKE ANY CORRECTIONS? (Y OR N) '
    +,2X,\)
     READ(*,618)ANST
618  FORMAT(A1)
C
C   MAKE CORRECTIONS OR CHANGES TO TEMPERATURE ENTRIES
C
     IF (ANST.EQ.'Y') THEN
42      CALL CLS
        WRITE(*,268)
        PRINT *,'  1.) INITIAL CHIP TEMPERATURE        ',IT,' ',UT
        PRINT *,'  2.) UPPER AMBIENT TEMPERATURE        ',UPRT,' ',UT
        PRINT *,'  3.) LOWER AMBIENT TEMPERATURE        ',LWRT,' ',UT
        PRINT *,'  4.) RIGHT AMBIENT TEMPERATURE        ',RT,' ',UT
        PRINT *,'  5.) LEFT AMBIENT TEMPERATURE        ',LT,' ',UT
        PRINT *,'  6.) FRONT AMBIENT TEMPERATURE        ',FT,' ',UT
        PRINT *,'  7.) REAR AMBIENT TEMPERATURE        ',BT,' ',UT
C
        WRITE(*,2682)
2682    FORMAT(//,'     WHICH TEMPERATURE WOULD YOU LIKE TO CHANGE?',/
```

```
     +,'    SELECT ZERO THROUGH SEVEN.  ',2X,\)
      READ(*,618)DELT
C
C  CORRECT INITIAL TEMPERATURE
      IF (DELT.EQ.'1') THEN
        CALL CLS
        WRITE(*,270)
270      FORMAT(//)
        PRINT *,'    INITIAL CHIP TEMPREATURE IS ',IT,' ',UT
        WRITE(*,271)UT
271      FORMAT(/,'     ENTER THE NEW VALUE.(',A1,')  ',2X,\)
        READ *,IT
C
C  CORRECT UPPER AMBIENT TEMPERATURE
      ELSEIF (DELT.EQ.'2') THEN
        CALL CLS
        WRITE(*,620)
620      FORMAT(//)
        PRINT *,'    UPPER AMBIENT TEMPERATURE IS ',UPRT,' ',UT
        WRITE(*,273)UT
273      FORMAT(/,'     ENTER THE NEW VALUE.(',A1,')  ',2X,\)
        READ *,UPRT
C
C  CORRECT LOWER AMBIENT TEMPERATURE
      ELSEIF (DELT.EQ.'3') THEN
        CALL CLS
        WRITE(*,623)
623      FORMAT(//)
        PRINT *,'    LOWER AMBIENT TEMPERATURE IS ',LWRT,' ',UT
        WRITE(*,275)UT
275      FORMAT(/,'     ENTER THE NEW VALUE.(',A1,')  ',2X,\)
        READ *,LWRT
C
C  CORRECT RIGHT AMBIENT TEMPERATURE
      ELSEIF (DELT.EQ.'4') THEN
        CALL CLS
        WRITE(*,626)
626      FORMAT(//)
        PRINT *,'    RIGHT AMBIENT TEMPERATURE IS ',RT,' ',UT
        WRITE(*,277)UT
277      FORMAT(/,'     ENTER THE NEW VALUE.(',A1,')  ',2X,\)
        READ *,RT
C
C  CORRECT LEFT AMBIENT TEMPERATURE
      ELSEIF (DELT.EQ.'5') THEN
        CALL CLS
        WRITE(*,629)
629      FORMAT(//)
        PRINT *,'    LEFT AMBIENT TEMPERATURE IS ',LT,' ',UT
        WRITE(*,279)UT
279      FORMAT(/,'     ENTER THE NEW VALUE.(',A1,')  ',2X,\)
        READ *,LT
C
C  CORRECT FRONT AMBIENT TEMPERATURE
      ELSEIF (DELT.EQ.'6') THEN
```

```
        CALL CLS
        WRITE(*,632)
632      FORMAT(//)
        PRINT *,'     FRONT AMBIENT TEMPERATURE IS ',FT,' ',UT
        WRITE(*,281)UT
281      FORMAT(','     ENTER THE NEW VALUE.(',A1,')  ',2X,\)
        READ *,FT
C
C  CORRECT REAR AMBIENT TEMPERATURE
      ELSEIF (DELT.EQ.'7') THEN
        CALL CLS
        WRITE(*,635)
635      FORMAT(///)
        PRINT *,'     REAR AMBIENT TEMPERATURE IS ',BT,' ',UT
        WRITE(*,284)UT
284      FORMAT(','     ENTER THE NEW VALUE.(',A1,')  ',2X,\)
        READ *,BT
C
C  MAKE NO CHANGES
      ELSEIF (DELT.EQ.'0') THEN
4201          CALL CLS
        WRITE(*,6320)
6320      FORMAT(///,'    YOU HAVE DECIDED TO MAKE NO CORRECTIONS!',/,
     +'    IS THIS CORRECT? (Y OR N)  ',2X,\)
        READ(*,618) ANSTL
        IF (ANSTL.EQ.'Y') THEN
          GOTO 51
        ELSEIF (ANSTL.EQ.'N') THEN
          GOTO 42
        ELSE
          GOTO 4201
        ENDIF
C
      ELSE
        GOTO 42
      ENDIF
C
C  ALLOW FOR ANOTHER REVIEW OF TEMPERATURE ENTRIES
C
48      CALL CLS
      WRITE(*,635)
      WRITE(*,285)
285    FORMAT(///,'    WOULD YOU LIKE TO MAKE ANY MORE CORRECTIONS',/,
     +'    OR REVIEW TEMPERATURE ENTRIES. (Y OR N)  ',2X,\)
      READ(*,618)ANSTL
      IF (ANSTL.EQ.'Y') THEN
        GOTO 42
      ELSEIF (ANSTL.EQ.'N') THEN
        GOTO 51
      ELSE
        GOTO 48
      ENDIF
C
    ELSEIF (ANSTL.EQ.'N') THEN
      GOTO 51
```

```
      ELSE
        GOTO 50
      ENDIF
C
C
C
C*****************************************************************************
C****************************HEAT INPUT***************************************
C*****************************************************************************
C
C   PROVIDE A CORRELATION BETWEEN NODE NUMBERRS AND MATRIX LOCATION
C
51    NUM = 1
      DO 60 I = 1,NPL/NWIDE
        DO 61 J = 1,NWIDE
          JH(NUM) = J
          NUM = NUM + 1
61      CONTINUE
60    CONTINUE
C
      NUMA = 1
      CH = 0
      H = 1
      DO 62 I = 1,NPL
        IH(NUMA) = H
        CH = CH + 1
        IF (CH.EQ.NWIDE) THEN
          H = H + 1
          CH = 0
        ENDIF
      NUMA = NUMA + 1
62    CONTINUE
C
C   PROVIDE ALTERNATIVE HEAT INPUT METHODS
C
      CALL CLS
      WRITE(*,290)
290   FORMAT('************************************************************
     +***************',/,
     +'**************************HEAT INPUT****************************
     +********',/,
     +'***************************************************************
     +********',//,
     +'     HEAT INPUT TO THE MICROCIRCUIT OCCURS ONLY ON THE UPPER',/,
     +'     SUBSTRATE SUFACE. HEAT INPUT IS ACCOMPLISHED BY ONE OF ',/,
     +'     THE FOLLOWING METHODS:',//,
     +'     1.) ENTER AS A TOTAL HEAT APPLIED TO THE CHIP.',/,
     +'     2.) ENTER AS AVERAGE HEAT PER UNIT AREA.',/,
     +'     3.) ENTER HEAT NODE BY NODE.',/,
     +'     4.) NO HEAT INPUT.',//,
     +'     PLEASE SELECT A NUMBER ONE THROUGH FOUR.  ',2X,\)
      READ(*,304)SELH
C
      IF (SELH.EQ.'1'.OR.SELH.EQ.'2'.OR.SELH.EQ.'3'.OR.SELH.EQ.'4') THEN
64      WRITE(*,291)SELH
```

```
291     FORMAT(//,'    YOU HAVE SELECTED NUMBER ',A1,' OF FOUR ALTERNAT
     +IVES. ',/,'    IS THIS THE CORRECT CHOICE? (Y OR N)  ',2X,\)
        READ(*,304)ANSH
C
        IF (ANSH.EQ.'Y') THEN
          GOTO 63
        ELSEIF (ANSH.EQ.'N') THEN
          GOTO 51
        ELSE
          CALL CLS
          WRITE(*,290)
          GOTO 64
63      ENDIF
C
      ELSE
        GOTO 51
      ENDIF
C
C   DETERMINE UNITS FOR HEAT INPUT
C
      IF ((ANSN.EQ.'E').AND.(SELH.EQ.'2')) THEN
        UH = 'Btu/(hr*in^2)'
        UAH = 'Btu/hr'
      ELSEIF ((ANSN.EQ.'E').AND.(SELH.EQ.'1'.OR.SELH.EQ.'3')) THEN
        UH = '(Btu) / (hr) '
      ELSEIF ((ANSN.EQ.'S').AND.(SELH.EQ.'2')) THEN
        UH = 'WATTS/(cm^2) '
        UAH = 'WATTS '
      ELSEIF ((ANSN.EQ.'S').AND.(SELH.EQ.'1'.OR.SELH.EQ.'3')) THEN
        UH = '    WATTS    '
      ENDIF
C
C
C   ALLOW FOR RE-SELECTION OF HEAT INPUT METHOD OR CONTINUE WITH
C   INITIAL SELECTION
C
C   CHOICE #1
C
      CALL CLS
      IF (SELH.EQ.'1') THEN
65      WRITE(*,292)UH
292     FORMAT(//,'    YOU HAVE SELECTED TO INPUT HEAT AS A TOTAL HEAT
     +',/,'    APPLIED TO THE SURFACE. ',//,
     +'    ENTER TOTAL HEAT APPLIED TO THE SURFACE (',A13,')  ',2X,\)
        READ *,THEAT
66      WRITE(*,293)
293     FORMAT(/,'    IS THIS THE CORRECT ENTRY (Y OR N)  ',2X,\)
        READ(*,304)ATH
C
C   MAKE HEAT ENTRY AND ALLOW FOR CORRECTION
C
        IF (ATH.EQ.'Y') THEN
          THPN = THEAT/NPL
          PRINT *
          PRINT *,'    TOTAL HEAT PER NODE IS ',THPN,' ',UH
```

```
C
C   FILL HEAT MATRIX WITH DESIRED VALUES
C
          DO 80 I = 1,NDEEP
            DO 81 J = 1,NWIDE
              HEAT(I,J) = THPN
81          CONTINUE
80        CONTINUE
C
        ELSEIF (ATH.EQ.'N') THEN
          CALL CLS
          GOTO 65
        ELSE
          CALL CLS
          WRITE(*,294)
294       FORMAT(////)
          PRINT *,'    TOTAL HEAT APPLIED TO THE SURFACE IS ',THEAT,'
     + ',UH
          GOTO 66
        ENDIF
C
C   CHOICE #2
C
      ELSEIF (SELH.EQ.'2') THEN
67      WRITE(*,295)UH
295     FORMAT(//,'    YOU HAVE SELECTED TO ENTER THE AVERAGE HEAT OV
     +ER THE',/,
     +'    UPPER SUBSTRATE SURFACE.',//,
     +'    ENTER THE DESIRED HEAT INPUT (',A13,'). ',2X,\)
        READ *,AHEAT
C
C   MAKE ENTRY AND ALLOW FOR CORRECTION
C
68      WRITE(*,296)
296     FORMAT(//,'    IS THIS THE CORRECT ENTRY. (Y OR N)  ',2X,\)
        READ(*,304)ANSHA
C
        IF (ANSHA.EQ.'Y') THEN
          THPN = AHEAT*SL*SW/NPL
          PRINT *
          PRINT *,'    TOTAL HEAT PER NODE IS ',THPN,' ',UAH
C
C   FILL HEAT MATRIX WITH DESIRED VALUES
C
        DO 82 I = 1,NDEEP
          DO 83 J = 1,NWIDE
            HEAT(I,J) = THPN
83        CONTINUE
82      CONTINUE
C
        ELSEIF (ANSHA.EQ.'N') THEN
          CALL CLS
          GOTO 67
        ELSE
          CALL CLS
```

```
          WRITE(*,294)
          PRINT *,'    AVERAGE HEAT OVER SUBSTRATE SURFACE IS ',AHEAT,
     +' ',UH
          GOTO 68
        ENDIF
C
C  CHOICE #3
C
       ELSEIF (SELI.EQ.'3') THEN
70      WRITE(*,297)
297      FORMAT(  ,'    YOU HAVE SELECTED TO ENTER HEAT NODALLY',//,
     +'  ENTER THE TOTAL NUMBER OF NODES DESIGNATED FOR HEAT INPUT.  '
     +,2X,\)
        READ *,TOTNOD
C
C  THIS IS NODE BY NODE. GET NUMBER OF ENTRIES THEN LOOP UNTIL ALL
C  ENTRIES HAVE BEEN MADE
C
C  TELL USER MAXIMUM ENTRIES POSSIBLE
C
        IF (TOTNOD.GT.NPL) THEN
          PRINT *
          PRINT *,'    MAXIMUM ENTRY IS ',NPL
          CALL CLS
          GOTO 70
        ENDIF
C
C  MAKE ENTRIES
C
        DO 71 I = 1,TOTNOD
          NC = 1
75         CALL CLS
          WRITE(*,402)NC,TOTNOD
402        FORMAT(/,'    THIS IS NUMBER ',I3,' OF ',I3,' ENTRIES')
          WRITE(*,298)
298        FORMAT(/,'    ENTER THE NODE NUMBER FOR HEAT INPUT.  ',2X,\)
          READ *,NN
          IF (NN.EQ.0.OR.NN.GT.NPL) THEN
            GOTO 75
          ENDIF
          WRITE(*,299)UH
299        FORMAT(/,'    ENTER HEAT INPUT (',A13,').  ',2X,\)
          READ *,NHEAT
          HEAT(IH(NN),JH(NN)) = NHEAT
71      CONTINUE
C
C  PROVIDE OPPORTUNITY FOR CORRECTIONS OR FURTHER ENTRIES
C
73      CALL CLS
        WRITE(*,401)TOTNOD
401      FORMAT( /,'    YOU HAVE MADE ',I3,' NODAL ENTRIES.')
        WRITE(*,400)
400      FORMAT(/,'    DO YOU WISH TO MAKE ANY MORE ENTRIES OR CORRECTIO
     +NS? (Y OR N)  ',2X,\)
        READ(*,304)AHN
```

```
C
      IF (AHN.EQ.'Y') THEN
        CALL CLS
        GOTO 70
      ELSEIF (AHN.EQ.'N') THEN
        GOTO 72
      ELSE
        GOTO 73
72     ENDIF
C
C
C  NO HEAT INPUT - CHOICE #4 - HEAT MATRIX STAYS INITIALIZED AT ZERO
C
    ELSEIF (SELH.EQ.'4') THEN
      GOTO 86
    ENDIF
C
86   CONTINUE
C
C
C*********************************************************************
C**********************DETERMINE COEFFICIENTS***********************
C*********************************************************************
C
C  DETERMINE INCREMENTAL MEASUREMENTS IN THE X AND Y DIRECTIONS
C
    DELX = SL/NDEEP
    DELY = SW/NWIDE
C
C*************GENERATE CONSTANTS FOR THE SUBSTRATE LAYER****************
C
C LEFT OR RIGHT EDGE TO OUTSIDE
    SYLR = 2 * KS * DELX * DELS / DELY
C
C FRONT OR BACK TO OUTSIDE
    SXFB = 2 * KS * DELY * DELS /DELX
C
C INNER MATRIX MOVEMENT IN THE Y DIRECTION
    SYY = KS * DELX * DELS / DELY
C
C INNER MATRIX MOVEMENT IN THE X DIRECTION
    SXX = KS * DELY * DELS / DELX
C
C SUBSTRATE TO TOP SURFACE
    SZT = 2 * KS * DELX *DELY / DELS
C
C SUBSTRATE TO EPOXY
    SZE = 2 * DELX * DELY / ((DELS/KS)+(DELE/KE))
C
C******************GENERATE CONSTANTS FOR EPOXY LAYER******************
C
C  LEFT OR RIGHT EDGE TO OUTSIDE
    EYLR = 2*KE*DELX*DELE/DELY
C
C  FRONT OR BACK EDGE TO OUTSIDE
```

```
      EXFB = 2*KE*DELY*DELE/DELX
C
C   INNER MATRIX MOVEMENT IN THE Y DIRECTION
      EYY = KE*DELX*DELE/DELY
C
C   EPOXY TO CARRIER
      EZC = 2*DELX*DELY/((DELE/KE)+(DELC/KC))
C
C   INNER MATRIX MOVEMENT IN THE X DIRECTION
      EXX = KE*DELE*DELY/DELX
C
C******************GENERATE CONSTANTS FOR INTERIOR CARRIER LAYER********
C
C   LEFT OR RIGHT EDGE TO OUTSIDE
      CYLR = 2*KC*DELX*DELC/DELY
C
C   INNER MATRIX MOVEMENT IN THE Y DIRECTION
      CYY = KC*DELX*DELC/DELY
C
C   INNER MATRIX MOVEMENT IN THE X DIRECTION
      CXX = KC*DELC*DELY/DELX
C
C   CARRIER TO BOTTOM OUTER EDGE
      CZB = 2*KC*DELX*DELY/DELC
C
C   CARRIER TO FRONT EDGE IF NO EAR EXISTS
      CXFB = 2*KC*DELY*DELC/DELX
C
C
C*******************************************************************
C****COEFFICIENTS FOR SUBSTRATE, EPOXY, AND INTERIOR CARRIER LAYERS*****
C*******************************************************************
C
C
      DO 90 I = 1,NPL
      N = 1
      IB = NPL + I
      ID = 2*NPL+I
C
C****************************CORNERS************************************
C
C
        IF ((IH(I).EQ.1.OR.IH(I).EQ.NDEEP).AND.(JH(I).EQ.1.OR.JH(I).EQ.
     +NWIDE)) THEN
C
C   DETERMINE CONNECTIONS FOR TOP LAYER
          IF (HEAT(IH(I),JH(I)).EQ.0.0) THEN
            NCON(I,N) = 6
          ELSE
            NCON(I,N) = 7
          ENDIF
C
C   CONNECTIONS FOR EPOXY LAYER
          NCON(IB,N) = 6
C
```

```
C
C
C   LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C       LEFT EDGE
         IF (JH(I).EQ.1) THEN
C       LEFT COEFFICIENT
            COEF(I,N) = SYLR
            COEF(IB,N) = EYLR
            N = N + 1
            NCON(I,N) = 7551
            NCON(IB,N) = 7551
C       RIGHT COEFFICIENT
            COEF(I,N) = SYY
            COEF(IB,N) = EYY
            N = N + 1
            NCON(I,N) = 10*(I+1)+1
            NCON(IB,N) = 10*(IB+1)+1
C       RIGHT EDGE
         ELSEIF (JH(I).EQ.NWIDE) THEN
C       LEFT COEFFICIENT
            COEF(I,N) = SYY
            COEF(IB,N) = EYY
            N = N + 1
            NCON(I,N) = 10*(I-1)+1
            NCON(IB,N) = 10*(IB-1)+1
C       RIGHT COEFFICIENT
            COEF(I,N) = SYLR
            COEF(IB,N) = EYLR
            N = N + 1
            NCON(I,N) = 7541
            NCON(IB,N) = 7541
         ENDIF
C
C   FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C       FRONT EDGE
         IF (IH(I).EQ.1) THEN
C       FRONT COEFFICIENT
            COEF(I,N) = SXFB
            COEF(IB,N) = EXFB
            N = N + 1
            NCON(I,N) = 7521
            NCON(IB,N) = 7521
C       BACK COEFFICIENT
            COEF(I,N) = SXX
            COEF(IB,N) = EXX
            N = N + 1
            NCON(I,N) = 10*(I+NWIDE)+1
            NCON(IB,N) = 10*(IB+NWIDE)+1
C       BACK EDGE
         ELSEIF (IH(I).EQ.NDEEP) THEN
C       FRONT COEFFICIENT
            COEF(I,N) = SXX
            COEF(IB,N) = EXX
            N = N + 1
            NCON(I,N) = 10*(I-NWIDE)+1
```

```
            NCON(IB,N) = 10*(IB-NWIDE)+1
C         BACK COEFFICIENT
            COEF(I,N) = SXFB
            COEF(IB,N) = EXFB
            N = N + 1
            NCON(I,N) = 7531
            NCON(IB,N) = 7531
          ENDIF
C
C   TOP COEFFICIENT
          COEF(I,N) = SZT
          COEF(IB,N) = SZE
          N = N + 1
          NCON(I,N) = 7511
          NCON(IB,N) = 10*I+1
C   BOTTOM COEFFICIENT
          COEF(I,N) = SZE
          COEF(IB,N) = EZC
          N = N + 1
          NCON(I,N) = 10*(I+NPL)+1
          NCON(IB,N) = 10*(IB+NPL)+1
C   HEAT INPUT
          IF (HEAT(IH(I),JH(I)).NE.0.0) THEN
            COEF(I,N) = HEAT(IH(I),JH(I))
            N = N + 1
            NCON(I,N) = 9991
          ENDIF
C
C
C***********FRONT AND BACK EDGES EXCLUDING CORNERS*********************
C
C
        ELSEIF ((IH(I).EQ.1.OR.IH(I).EQ.NDEEP).AND.(JH(I).NE.1.OR.JH(I)
     +.NE.NWIDE)) THEN
C
C   DETERMINE NUMBER OF CONNECTIONS FOR SUBSTRATE LAYER
          IF (HEAT(IH(I),JH(I)).EQ.0.0) THEN
            NCON(I,N) = 6
          ELSE
            NCON(I,N) = 7
          ENDIF
C
C   DETERMINE NUMBER OF CONNECTIONS FOR EPOXY LAYER
          NCON(IB,N) = 6
C
C   LEFT COEFFICIENT
          COEF(I,N) = SYY
          COEF(IB,N) = EYY
          N = N + 1
          NCON(I,N) = 10*(I-1)+1
          NCON(IB,N) = 10*(IB-1)+1
C   RIGHT COEFFICIENT
          COEF(I,N) = SYY
          COEF(IB,N) = EYY
          N = N + 1
```

```
        NCON(I,N) = 10*(I+1)+1
        NCON(IB,N) = 10*(IB+1)+1
C
C  FRONT AND BACK COEFFICIENTS DEPENDENT ON WHICH EDGE
C      FRONT EDGE
        IF (JH(I).EQ.1) THEN
C      FRONT COEFFICIENT
          COEF(I,N) = SXFB
          COEF(IB,N) = EXFB
          N = N + 1
          NCON(I,N) = 7521
          NCON(IB,N) = 7521
C      BACK COEFFICIENT
          COEF(I,N) = SXX
          COEF(IB,N) = EXX
          N = N + 1
          NCON(I,N) = 10*(I+NWIDE)+1
          NCON(IB,N) = 10*(IB+NWIDE)+1
C      BACK EDGE
        ELSEIF (JH(I).EQ.NDEEP) THEN
C      FRONT COEFFICIENT
          COEF(I,N) = SXX
          COEF(IB,N) = EXX
          N = N + 1
          NCON(I,N) = 10*(I-NWIDE)+1
          NCON(IB,N) = 10*(IB-NWIDE)+1
C      BACK COEFFICIENT
          COEF(I,N) = SXFB
          COEF(IB,N) = EXFB
          N = N + 1
          NCON(I,N) = 7531
          NCON(IB,N) = 7531
        ENDIF
C
C  TOP COEFFICIENT
        COEF(I,N) = SZT
        COEF(IB,N) = SZE
        N = N + 1
        NCON(I,N) = 7511
        NCON(IB,N) = 10*I+1
C
C  BOTTOM COEFFICIENT
        COEF(I,N) = SZE
        COEF(IB,N) = EZC
        N = N + 1
        NCON(I,N) = 10*(I+NPL)+1
        NCON(IB,N) = 10*(IB+NPL)+1
C
C  HEAT INPUT
        IF (HEAT(IH(I),JH(I)).NE.0.0) THEN
          COEF(I,N) = HEAT(IH(I),JH(I))
          N = N + 1
          NCON(I,N) = 9991
        ENDIF
C
```

```
C
C***********LEFT AND RIGHT EDGES EXCLUDING CORNERS*********************
C
C
      ELSEIF ((JH(I).EQ.1.OR.JH(I).EQ.NWIDE).AND.(IH(I).NE.1.OR.IH(I)
     +.NE.NDEEP)) THEN
C
C  DETERMINE NUMBER OF CONNECTIONS FOR SUBSTRATE LAYER
         IF (HEAT(IH(I),JH(I)).NE.0.0) THEN
           NCON(I,N) = 7
         ELSE
           NCON(I,N) = 6
         ENDIF
C
C  DETERMINE THE NUMBER OF CONNECTIONS FOR EPOXY AND CARRIER LAYERS
         NCON(IB,N) = 6
         NCON(ID,N) = 6
C
C
C  LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C       LEFT EDGE
         IF (JH(I).EQ.1) THEN
C       LEFT COEFFICIENT
           COEF(I,N) = SYLR
           COEF(IB,N) = EYLR
           COEF(ID,N) = CYLR
           N = N + 1
           NCON(I,N) = 7551
           NCON(IB,N) = 7551
           NCON(ID,N) = 7551
C       RIGHT COEFFICIENT
           COEF(I,N) = SYY
           COEF(IB,N) = EYY
           COEF(ID,N) = CYY
           N = N + 1
           NCON(I,N) = 10*(I+1)+1
           NCON(IB,N) = 10*(IB+1)+1
           NCON(ID,N) = 10*(ID+1)+1
C       RIGHT EDGE
         ELSEIF (JH(I).EQ.NWIDE) THEN
C       LEFT COEFFICIENT
           COEF(I,N) = SYY
           COEF(IB,N) = EYY
           COEF(ID,N) = CYY
           N = N + 1
           NCON(I,N) = 10*(I-1)+1
           NCON(IB,N) = 10*(IB-1)+1
           NCON(ID,N) = 10*(ID-1)+1
C       RIGHT COEFFICIENT
           COEF(I,N) = SYLR
           COEF(IB,N) = EYLR
           COEF(ID,N) = CYLR
           N = N + 1
           NCON(I,N) = 7541
           NCON(IB,N) = 7541
```

```
            NCON(ID,N) = 7541
          ENDIF
C
C     FRONT COEFFICIENT
          COEF(I,N) = SXX
          COEF(IB,N) = EXX
          COEF(ID,N) = CXX
          N = N + 1
          NCON(I,N) = 10*(I-NWIDE)+1
          NCON(IB,N) = 10*(IB-NWIDE)+1
          NCON(ID,N) = 10*(ID-NWIDE)+1
C
C     BACK COEFFICIENT
          COEF(I,N) = SXX
          COEF(IB,N) = EXX
          COEF(ID,N) = CXX
          N = N + 1
          NCON(I,N) = 10*(I+NWIDE)+1
          NCON(IB,N) = 10*(IB+NWIDE)+1
          NCON(ID,N) = 10*(ID+NWIDE)+1
C
C     TOP COEFFICIENT
          COEF(I,N) = SZT
          COEF(IB,N) = SZE
          COEF(ID,N) = EZC
          N = N + 1
          NCON(I,N) = 7511
          NCON(IB,N) = 10*I+1
          NCON(ID,N) = 10*IB+1
C
C     BOTTOM COEFFICIENT
          COEF(I,N) = SZE
          COEF(IB,N) = EZC
          COEF(ID,N) = CZB
          N = N + 1
          NCON(I,N) = 10*(I+NPL)+1
          NCON(IB,N) = 10*(IB+NPL)+1
          NCON(ID,N) = 7561
C
C     HEAT INPUT
          IF (HEAT(IH(I),JH(I)).NE.0.0) THEN
            COEF(I,N) = HEAT(IH(I),JH(I))
            N = N + 1
            NCON(I,N) = 9991
          ENDIF
C
C
C******DETERMINE COEFFICIENTS FOR ALL NODES NOT TOUCHING AN EDGE********
C
C
        ELSEIF ((IH(I).NE.1.OR.IH(I).NE.NDEEP).AND.(JH(I).NE.1.OR.JH(I)
     +.NE.NWIDE)) THEN
C
C   DETERMINE NUMBER OF CONNECTIONS FOR SUBSTRATE LAYER
          IF (HEAT(IH(I),JH(I)).NE.0.0) THEN
```

```
            NCON(I,N) = 7
          ELSE
            NCON(I,N) = 6
          ENDIF
C
C   DETERMINE NUMBER OF CONNECTIONS FOR EPOXY AND CARRIER LAYERS
          NCON(IB,N) = 6
          NCON(ID,N) = 6
C
C    LEFT COEFFICIENT
          COEF(I,N) = SYY
          COEF(IB,N) = EYY
          COEF(ID,N) = CYY
          N = N + 1
          NCON(I,N) = 10*(I-1)+1
          NCON(IB,N) = 10*(IB-1)+1
          NCON(ID,N) = 10*(ID-1)+1
C
C    RIGHT COEFFICIENT
          COEF(I,N) = SYY
          COEF(IB,N) = EYY
          COEF(ID,N) = CYY
          N = N + 1
          NCON(I,N) = 10*(I+1)+1
          NCON(IB,N) = 10*(IB+1)+1
          NCON(ID,N) = 10*(ID+1)+1
C
C    FRONT COEFFICIENT
          COEF(I,N) = SXX
          COEF(IB,N) = EXX
          COEF(ID,N) = CXX
          N = N + 1
          NCON(I,N) = 10*(I-NWIDE)+1
          NCON(IB,N) = 10*(IB-NWIDE)+1
          NCON(ID,N) = 10*(ID-NWIDE)+1
C
C    BACK COEFFICIENT
          COEF(I,N) = SXX
          COEF(IB,N) = EXX
          COEF(ID,N) = CXX
          N = N + 1
          NCON(I,N) = 10*(I+NWIDE)+1
          NCON(IB,N) = 10*(IB+NWIDE)+1
          NCON(ID,N) = 10*(ID+NWIDE)+1
C
C    TOP COEFFICIENT
          COEF(I,N) = SZT
          COEF(IB,N) = SZE
          COEF(ID,N) = EZC
          N = N + 1
          NCON(I,N) = 7511
          NCON(IB,N) = 10*I+1
          NCON(ID,N) = 10*IB+1
C
C    BOTTOM COEFFICIENT
```

```
          COEF(I,N) = SZE
          COEF(IB,N) = EZC
          COEF(ID,N) = CZB
          N = N + 1
          NCON(I,N) = 10*(I+NPL)+1
          NCON(IB,N) = 10*(IB+NPL)+1
          NCON(ID,N) = 7561
C
C     HEAT INPUT
          IF (HEAT(IH(I),JH(I)).NE.0.0) THEN
            COEF(I,N) = HEAT(IH(I),JH(I))
            N = N + 1
            NCON(I,N) = 9991
          ENDIF
C
        ENDIF
90    CONTINUE
C
C
C***********************************************************************
C********COEFFICIENTS FOR CARRIER LAYER FRONT AND BACK EDGES************
C***********************************************************************
C
C   DETERMINE CONSTANTS FOR CARRIER COEFFICIENTS
C
C   EAR SIZE
      EAR = (CL-SL)/2
C
C***********************************************************************
C***************EAR COEFFICIENTS IF NECCESARY***************************
C***********************************************************************
C
      IF (EAR.NE.0.0) THEN
C
C   DETERMINE THE NECCESARY CONSTANTS
C   CONSTANTS FOR EAR NODES
C   LEFT OR RIGHT TO EXTERIOR
      CYLRE = 2*KC*EAR*DELC/DELY
C
C   LEFT OR RIGHT TO INTERIOR
      CYYE = EAR*KC*DELC/DELY
C
C   FRONT OR BACK TO EXTERIOR
      CXFBE = 2*DELY*KC*DELC/EAR
C
C   FRONT OR BACK TO INTERIOR
      CXXE = DELY*DELC*KC*2/(EAR+DELX)
C
C   TOP OR BOTTOM
      CZEAR = KC*EAR*DELY*2/DELC
C
C   DETERMINE EAR COEFFICIENTS
        DO 100 I = 1,2*NWIDE
          IE = 3*NPL+I
          N = 1
```

```
C   EAR NODE CONNECTIONS
          NCON(IE,N) = 6
C
C   CORNERS
          IF ((IH(I).EQ.1.OR.IH(I).EQ.2).AND.(JH(I).EQ.1.OR.JH(I).EQ.
     +NWIDE)) THEN
C   LEFT EDGE
            IF (JH(I).EQ.1) THEN
C       LEFT COEFFICIENT
              COEF(IE,N) = CYLRE
              N = N + 1
              NCON(IE,N) = 7551
C       RIGHT COEFFICIENT
              COEF(IE,N) = CYYE
              N = N + 1
              NCON(IE,N) = 10*(IE+1)+1
            ELSEIF (JH(I).EQ.NWIDE) THEN
C   RIGHT EDGE
C       LEFT COEFFICIENT
              COEF(IE,N) = CYYE
              N = N + 1
              NCON(IE,N) = 10*(IE-1)+1
C       RIGHT COEFFICIENT
              COEF(IE,N) = CYLRE
              N = N + 1
              NCON(IE,N) = 7541
            ENDIF
C
C   FRONT, BACK AND TOP COEFFICIENTS
C   FRONT EDGE
            IF (IH(I).EQ.1) THEN
C       FRONT COEFFICIENT
              COEF(IE,N) = CXFBE
              N = N + 1
              NCON(IE,N) = 75[illegible]1
C       BACK COEFFICIENT
              COEF(IE,N) = CXXE
              N = N + 1
              NCON(IE,N) = 10*(2*NPL+JH(I))+1
C       TOP COEFFICIENT
              COEF(IE,N) = CZEAR
              N = N + 1
              NCON(IE,N) = 7521
C   BACK EDGE
            ELSEIF (IH(I).EQ.2) THEN
C       FRONT COEFFICIENT
              COEF(IE,N) = CXXE
              N = N + 1
              NCON(IE,N) = 10*(3*NPL-NWIDE+JH(I))+1
C       BACK COEFFICIENT
              COEF(IE,N) = CXFBE
              N = N + 1
              NCON(IE,N) = 7531
C       TOP COEFFICIENT
              COEF(IE,N) = CZEAR
```

```
          N = N + 1
          NCON(IE,N) = 7531
        ENDIF
C
C  BOTTOM COEFFICIENT
          COEF(IE,N) = CZEAR
          N = N + 1
          NCON(IE,N) = 7561
C
C  EAR NODES ON FRONT EDGE EXCLUDING CORNERS
       ELSEIF ((IH(I).EQ.1).AND.(JH(I).NE.1.OR.JH(I).NE.NWIDE))THEN
C      LEFT COEFFICIENT
          COEF(IE,N) = CYYE
          N = N + 1
          NCON(IE,N) = 10*(3*NPL+JH(I)-1)+1
C      RIGHT EDGE
          COEF(IE,N) = CYYE
          N = N + 1
          NCON(IE,N) = 10*(3*NPL+JH(I)+1)+1
C      FRONT COEFFICIENT
          COEF(IE,N) = CXFBE
          N = N + 1
          NCON(IE,N) = 7521
C      BACK COEFFICIENT
          COEF(IE,N) = CXXE
          N = N + 1
          NCON(IE,N) = 10*(2*NPL+JH(I))+1
C      TOP COEFFICIENT
          COEF(IE,N) = CZEAR
          N = N + 1
          NCON(IE,N) = 7521
C       BOTTOM COEFFICIENT
          COEF(IE,N) = CZEAR
          N = N + 1
          NCON(IE,N) = 7561
C
C  EAR NODES ON BACK EDGE EXCLUDING CORNERS
       ELSEIF ((IH(I).EQ.2).AND.(JH(I).NE.1.OR.JH(I).NE.NWIDE))THEN
C      LEFT COEFFICIENT
          COEF(IE,N) = CYYE
          N = N + 1
          NCON(IE,N) = 10*(3*NPL+NWIDE+JH(I)-1)+1
C      RIGHT COEFFICIENT
          COEF(IE,N) = CYYE
          N = N + 1
          NCON(IE,N) = 10*(3*NPL+NWIDE+JH(I)+1)+1
C      FRONT COEFFICIENT
          COEF(IE,N) = CXXE
          N = N + 1
          NCON(IE,N) = 10*(3*NPL-NWIDE+JH(I))+1
C      BACK COEFFICIENT
          COEF(IE,N) = CXFBE
          N = N + 1
          NCON(IE,N) = 7531
C      TOP COEFFICIENT
```

```
          COEF(IE,N) = CZEAR
          N = N + 1
          NCON(IE,N) = 7531
C       BOTTOM COEFFICIENT
          COEF(IE,N) = CZEAR
          N = N + 1
          NCON(IE,N) = 7561
        ENDIF
100     CONTINUE
C
C
C
C*****FRONT AND BACK EDGE COEFFICIENTS IF AN EAR EXISTS*****************
C
C
C
      DO 101 I = 1,NPL
        N = 1
        IC = 2*NPL + I
C
C
C**********************CORNERS*****************************************
C
        IF ((IH(I).EQ.1.OR.IH(I).EQ.NDEEP).AND.(JH(I).EQ.1.OR.JH(I)
     +.EQ.NWIDE)) THEN
C
C   NUMBER OF CONNECTIONS PER NODE
        NCON(IC,N) = 6
C
C   LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C       LEFT EDGE
          IF (JH(I).EQ.1) THEN
C       LEFT COEFFICIENT
            COEF(IC,N) = CYLR
            N = N + 1
            NCON(IC,N) = 7551
C       RIGHT COEFFICIENT
            COEF(IC,N) = CYY
            N = N + 1
            NCON(IC,N) = 10*(IC+1)+1
C       RIGHT EDGE
          ELSEIF (JH(I).EQ.NWIDE) THEN
C       LEFT COEFFICIENT
            COEF(IC,N) = CYY
            N = N + 1
            NCON(IC,N) = 10*(IC-1)+1
C       RIGHT COEFFICIENT
            COEF(IC,N) = CYLR
            N = N + 1
            NCON(IC,N) = 7541
          ENDIF
C
C   FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C       FRONT EDGE
          IF (IH(I).EQ.1) THEN
```

```
C       FRONT COEFFICIENT
              COEF(IC,N) = CXXE
              N = N + 1
              NCON(IC,N) = 10*(3*NPL+JH(I))+1
C       BACK COEFFICIENT
              COEF(IC,N) = CXX
              N = N + 1
              NCON(IC,N) = 10*(IC+NWIDE)+1
C       BACK EDGE
           ELSEIF (IH(I).EQ.NDEEP) THEN
C       FRONT COEFFICIENT
              COEF(IC,N) = CXX
              N = N + 1
              NCON(IC,N) = 10*(IC-NWIDE)+1
C       BACK COEFFICIENT
              COEF(IC,N) = CXXE
              N = N + 1
              NCON(IC,N) = 10*(3*NPL+NWIDE+JH(I))+1
           ENDIF
C
C       TOP COEFFICIENTS
           COEF(IC,N) = EZC
           N = N + 1
           NCON(IC,N) = 10*(IC-NPL)+1
C       BOTTOM COEFFICIENT
           COEF(IC,N) = CZB
           N = N + 1
           NCON(IC,N) = 7561
C
C
C*********************FRONT AND BACK EDGES EXCLUDING CORNERS************
C
C
         ELSEIF ((IH(I).EQ.1.OR.IH(I).EQ.NDEEP).AND.(JH(I).NE.1.OR.
     +JH(I).NE.NWIDE)) THEN
C
C         NUMBER OF CONNECTIONS PER NODE
         NCON(IC,N) = 6
C
C       LEFT COEFFICIENT
           COEF(IC,N) = CYY
           N = N + 1
           NCON(IC,N) = 10*(IC-1)+1
C
C       RIGHT COEFFICIENT
           COEF(IC,N) = CYY
           N = N + 1
           NCON(IC,N) = 10*(IC+1)+1
C
C       FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C       FRONT EDGE
           IF (IH(I).EQ.1) THEN
C       FRONT COEFFICIENT
              COEF(IC,N) = CXFBE
              N = N + 1
```

```
           NCON(IC,N) = 10*(3*NPL+JH(I))+1
C      BACK COEFFICIENT
           COEF(IC,N) = CXX
           N = N + 1
           NCON(IC,N) = 10*(IC+NWIDE)+1
C      BACK EDGE
         ELSEIF (IH(I).EQ.NDEEP) THEN
C      FRONT COEFFICIENT
           COEF(IC,N) = CXX
           N = N + 1
           NCON(IC,N) = 10*(IC-NWIDE)+1
C      BACK COEFFICIENT
           COEF(IC,N) = CXFBE
           N = N + 1
           NCON(IC,N) = 10*(3*NPL+NWIDE+JH(I))+1
         ENDIF
C
C      TOP COEFFICIENT
         COEF(IC,N) = EZC
         N = N + 1
         NCON(IC,N) = 10*(IC-NPL)+1
C
C      BOTTOM COEFFICIENT
         COEF(IC,N) = CZB
         N = N + 1
         NCON(IC,N) = 7561
        ENDIF
101    CONTINUE
C
C**********************************************************************
C*****FRONT AND BACK EDGE CARRIER COEFFICIENT IF NO EAR*****************
C**********************************************************************
C
     ELSEIF (EAR.EQ.0.0) THEN
       DO 102 I = 1,NPL
         N = 1
         IC = 2*NPL+I
C
C********************CORNERS****************************************
C
         IF ((IH(I).EQ.1.OR.IH(I).EQ.NDEEP).AND.(JH(I).EQ.1.OR.JH(I).
    +EQ.NWIDE)) THEN
C
C   NUMBER OF CONNECTIONS PER NODE
         NCON(IC,N) = 6
C
C   LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C      LEFT EDGE
         IF (JH(I).EQ.1) THEN
C      LEFT COEFFICIENT
           COEF(IC,N) = CYLR
           N = N + 1
           NCON(IC,N) = 7551
C      RIGHT COEFFICIENT
           COEF(IC,N) = CYY
```

```
            N = N + 1
            NCON(IC,N) = 10*(IC+1)+1
C       RIGHT EDGE
          ELSEIF (JH(I).EQ.NWIDE) THEN
C       LEFT COEFFICIENT
            COEF(IC,N) = CYY
            N = N + 1
            NCON(IC,N) = 10*(IC-1)+1
C       RIGHT COEFFICIENT
            COEF(IC,N) = CYLR
            N = N + 1
            NCON(IC,N) = 7541
          ENDIF
C
C   FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C       FRONT EDGE
          IF (IH(I).EQ.1) THEN
C       FRONT COEFFICIENT
            COEF(IC,N) = CXFB
            N = N + 1
            NCON(IC,N) = 7521
C       BACK COEFFICIENT
            COEF(IC,N) = CXX
            N = N + 1
            NCON(IC,N) = 10*(IC+NWIDE)+1
C       BACK EDGE
          ELSEIF (IH(I).EQ.NDEEP) THEN
C       FRONT COEFFICIENT
            COEF(IC,N) = CXX
            N = N + 1
            NCON(IC,N) = 10*(IC-NWIDE)+1
C       BACK COEFFICIENT
            COEF(IC,N) = CXFB
            N = N + 1
            NCON(IC,N) = 7531
          ENDIF
C
C       TOP COEFFICIENT
          COEF(IC,N) = EZC
          N = N + 1
          NCON(IC,N) = 10*(IC-NPL)+1
C       BOTTOM COEFFICIENT
          COEF(IC,N) = CZB
          N = N + 1
          NCON(IC,N) = 7561
C
C
C*********FRONT AND BACK EDGES EXCLUDING CORNERS**********************
C
        ELSEIF ((IH(I).EQ.1.OR.IH(I).EQ.NDEEP).AND.(JH(I).NE.1.OR.
     +JH(I).NE.NWIDE)) THEN
C   NUMBER OF CONNECTIONS PER NODE
          NCON(IC,N) = 6
C
C       LEFT COEFFICIENT
```

```
          COEF(IC,N) = CYY
          N = N + 1
          NCON(IC,N) = 10*(IC-1)+1
C       RIGHT COEFFICIENT
          COEF(IC,N) = CYY
          N = N + 1
          NCON(IC,N) = 10*(IC+1)+1
C
C       FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C       FRONT EDGE
          IF (IH(I).EQ.1) THEN
C       FRONT COEFFICIENT
            COEF(IC,N) = CXFB
            N = N + 1
            NCON(IC,N) = 7521
C       BACK COEFFICIENT
            COEF(IC,N) = CXX
            N = N + 1
            NCON(IC,N) = 10*(IC+NWIDE)+1
C       BACK EDGE
          ELSEIF (IH(I).EQ.NDEEP) THEN
C       FRONT COEFFICIENT
            COEF(IC,N) = CXX
            N = N + 1
            NCON(IC,N) = 10*(IC-NWIDE)+1
C       BACK COEFFICIENT
            COEF(IC,N) = CXFB
            N = N + 1
            NCON(IC,N) = 7531
          ENDIF
C
C       TOP COEFFICIENT
          COEF(IC,N) = EZC
          N = N + 1
          NCON(IC,N) = 10*(IC-NPL)+1
C       BOTTOM COEFFICIENT
          COEF(IC,N) = CZB
          N = N + 1
          NCON(IC,N) = 7561
C
        ENDIF
102     CONTINUE
      ENDIF
C
C   GENERATE DATA FILE VALUES
C   TOTAL NODES FOR THIS ASPECT RATIO
      IF (EAR.EQ.0.0) THEN
        COUNT = 3*NPL
      ELSE
        COUNT = 3*NPL+2*NWIDE
      ENDIF
C
C   NUMBER OF CONSTANT TEMPERATURE INPUTS
      CONTEMP = 6
C
```

```
C   DUMMY VARIABLE
     ZER = 0
C
C   UNITS TO BE USED
     IF (UNITS.EQ.'E') THEN
       USEL = 1
     ELSE
       USEL = 2
     ENDIF
C
C   PROBLEM CAPABILITY LINE
C   MAXIMUM NODES
     NMAX = 750
C   MAXIMUM CONSTANT TEMPERATURES
     TMAX = 50
C   NUMBER OF HEATERS
     HTRS = 6
C   DATA SETS REQUIRED
     D1 = 2
     D2 = 4
     D3 = 6
     D4 = 0
     D5 = 0
     D6 = 0
     D7 = 0
C
C   ACCURACY LINE
C
C  ACCURACY BETWEEN ITERATIONS
     ACC = 0.05
C   DAMPING VALUE
     DAMP = 0.66667
C   MAXIMUM ITERATIONS
     MAXIT = 12
C   CONVERGENCE FACTOR
     CONFAC = 0.8
C
C
C
C  CREATE DATA FILE
     OPEN (3,FILE=NAME,FORM='FORMATTED',ACCESS='DIRECT',RECL=108
    +,STATUS='NEW')
C
C   LINE 1, TITLE
     WRITE(3,909) DATAF
909  FORMAT(1X,A79)
C
C   LINE 2, PROBLEM DATA
     WRITE(3,908) COUNT,CONTEMP,ZER,ZER,ZER,ZER,ZER ZER,USEL
908  FORMAT(2X,9(I3,5X))
C
C   ANALYZER CONTROL LINE
     WRITE(3,907) ZER,ZER,ZER
907  FORMAT(2X,3(I3,5X))
C
```

```
C   PROBLEM CAPABILITY LINE
      WRITE(3,908) NMAX,TMAX,HTRS,D1,D2,D3,D4,D5,D6,D7
C
C   ACCURACY LINE
      WRITE(3,905) ACC,DAMP,MAXIT,CONFAC,IT
905   FORMAT(1X,2(F9.7,1X),14X,I2,1X,F9.7,1X,F9.5)
C
C
C   CONSTANT TEMPERATURE LINE
      WRITE(3,906) UPRT,FT,BT,RT,LT,LWRT
906   FORMAT(1X,6(F12.9,1X))
C
C   COEFFICIENT EQUATIONS
      DO 112 I=1,COUNT
        WRITE(3,910) (NCON(I,J),J=1,8)
910     FORMAT(I4,3X,7(I4,8X))
        WRITE(3,911) (COEF(I,N),N=1,7)
911     FORMAT(7(F9.3,3X))
112   CONTINUE
C
C
C
      CALL CLS
      WRITE(*,999) NAME
999   FORMAT(///,'   THE OUTPUT DATA HAS BEEN PLACED IN A FILE NAMED '
     +,A6)
      END
```

LIST OF REFERENCES

1. Kraus,A.D.,and Bar-Cohen,A.,*Thermal Analysis and Control of Electronic Equipment*,pp.1-97,555-568,Hemisphere Publishing Corporation, 1983.

2. Arsenault,J.E.,and Roberts,J.A.,*Reliability and Maintainability of Electronic Systems*,pp.162-178,Computer Science Press,Inc.,1980.

3. Keonjian,E.,*Microelectronics*,pp.71-75,McGraw-Hill Book Company, Inc.,1963.

4. Sloan,J.L.,*Design and Packaging of Electronic Equipment*,pp.24-205, Van Nostrand-Reinhold Company,Inc.,1985.

5. Sedra,A.S.,and Smith,K.C.,*Microelectronic Circuits*,pp.564-591, Holt,Rinehart and Winston,Inc.,1987.

6. Obert,E.F.,and Young,R.L.,*Elements of Thermodynamics and Heat Transfer*,pp.370-434,McGraw-Hill Book Company,Inc.,1962.

7. Özişik,M.N.,*Heat Conduction*,pp.1-25,471-522,John Wiley and Sons,Inc.,1980.

8. Myers,G.E.,*Analytical Methods in Conduction Heat Transfer*,pp.1-24, 233-315,McGraw-Hill Book Company,Inc.,1971.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 52 2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Commander 1
   Naval Space Command
   Attn: Code N152
   Dahlgren, Virginia 22448

4. United States Space Command 1
   Attn: Technical Library
   Peterson AFB, Colorado 80914

5. Director 1
   Navy Space Systems Division (Op-943)
   Washington, DC 20350-2000

6. Space Systems Academic Group, Code SP 1
   Naval Postgraduate School
   Monterey, California 93943-5000

7. Department Chairman, Code EC 1
   Department of Electrical & Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943-5000

8. Prof. Allan D. Kraus, Code EC/KS 1
   Department of Electrical & Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943-5000

9. Prof. Sherif Michael, Code EC/MI 1
   Department of Electrical & Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943-5000